6.036: Introduction to Machine Learning

Cambridge MA elections: early voting now! Election Day Tues Nov 2

Lecture start: Tuesdays 9:35am

Who's talking? Prof. Tamara Broderick

Questions? Ask on Piazza: "lecture (week) 7" folder

Materials: slides, video will all be available on Canvas

Live Zoom feed: https://mit.zoom.us/j/94238622313

Last Time(s)

I. Neural nets: hypothesis class

II. Neurals nets: function graph representation

Today's Plan

I. More activation functions

II. Derivatives in neural nets (for SGD or GD)

Problem setup

Problem setup

1. Choose a hypothesis class.

Problem setup

1. Choose a hypothesis class.

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.

Problem setup

1. Choose a hypothesis class.

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.

3. Learn the parameters. E.g. gradient descent or SGD

Problem setup

1. Choose a hypothesis class.

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.
3. Learn the parameters. E.g. gradient descent or SGD
4. Predict on new data using these parameters

Problem setup

- # layer ℓ inputs $m^{(\ell)}$
- # layer ℓ outputs $n^{(\ell)}$

1. Choose a hypothesis class. E.g.,

$$h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$$

   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.
3. Learn the parameters. E.g. gradient descent or SGD
4. Predict on new data using these parameters

Problem setup

- # layer ℓ inputs $m^{(\ell)}$
- # layer ℓ outputs $n^{(\ell)}$

1. Choose a hypothesis class. E.g.,

$$h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$$

   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x +$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

$n^{(\ell)}$ is NOT the number of data points

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.
3. Learn the parameters. E.g. gradient descent or SGD
4. Predict on new data using these parameters

Problem setup

- # layer ℓ inputs $m^{(\ell)}$
- # layer ℓ outputs $n^{(\ell)}$

1. Choose a hypothesis class. E.g.,

$$h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$$

   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.
3. Learn the parameters. E.g. gradient descent or SGD
4. Predict on new data using these parameters

Issues:

Problem setup

- # layer ℓ inputs $m^{(\ell)}$
- # layer ℓ outputs $n^{(\ell)}$

1. Choose a hypothesis class. E.g.,

$$h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$$

   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.
3. Learn the parameters. E.g. gradient descent or SGD
4. Predict on new data using these parameters

Issues:

- What if I want to do regression or use NLL loss?

Problem setup

- # layer ℓ inputs $m^{(\ell)}$
- # layer ℓ outputs $n^{(\ell)}$

1. Choose a hypothesis class. E.g.,

$$h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$$

   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.
3. Learn the parameters. E.g. gradient descent or SGD
4. Predict on new data using these parameters

Issues:

- What if I want to do regression or use NLL loss?
- Derivatives (of loss with respect to parameters) are zero (or undefined) if we use step function activation

Problem setup

- # layer ℓ inputs $m^{(\ell)}$
- # layer ℓ outputs $n^{(\ell)}$

1. Choose a hypothesis class. E.g.,

$$h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$$

   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.
3. Learn the parameters. E.g. gradient descent or SGD
4. Predict on new data using these parameters

Issues:

- What if I want to do regression or use NLL loss?
- Derivatives (of loss with respect to parameters) are zero (or undefined) if we use step function activation
  - So (S)GD won't do what we want

Problem setup

- # layer ℓ inputs $m^{(\ell)}$
- # layer ℓ outputs $n^{(\ell)}$

1. Choose a hypothesis class. E.g.,

$$h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$$

   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

2. Choose a loss. (E.g. for classification: 0-1 loss, asymmetric, NLL.) Set up an objective.
3. Learn the parameters. E.g. gradient descent or SGD
4. Predict on new data using these parameters

Issues:

- What if I want to do regression or use NLL loss?
- Derivatives (of loss with respect to parameters) are zero (or undefined) if we use step function activation
  - So (S)GD won't do what we want
- How to compute the derivatives in (S)GD?

Different activation functions

Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

- What if I want to do regression?

Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

- What if I want to do regression? $f^{(2)}(z) = z$

Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

- What if I want to do regression? $f^{(2)}(z) = z$
- What if I want to use NLL loss?

Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

- What if I want to do regression? $f^{(2)}(z) = z$
- What if I want to use NLL loss? $f^{(2)}(z) = \sigma(z)$

Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

- What if I want to do regression? $f^{(2)}(z) = z$
- What if I want to use NLL loss? $f^{(2)}(z) = \sigma(z)$
- Need non-zero derivatives for (S)GD:

Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

- What if I want to do regression? $f^{(2)}(z) = z$
- What if I want to use NLL loss? $f^{(2)}(z) = \sigma(z)$
- Need non-zero derivatives for (S)GD: Above

Different activation functions

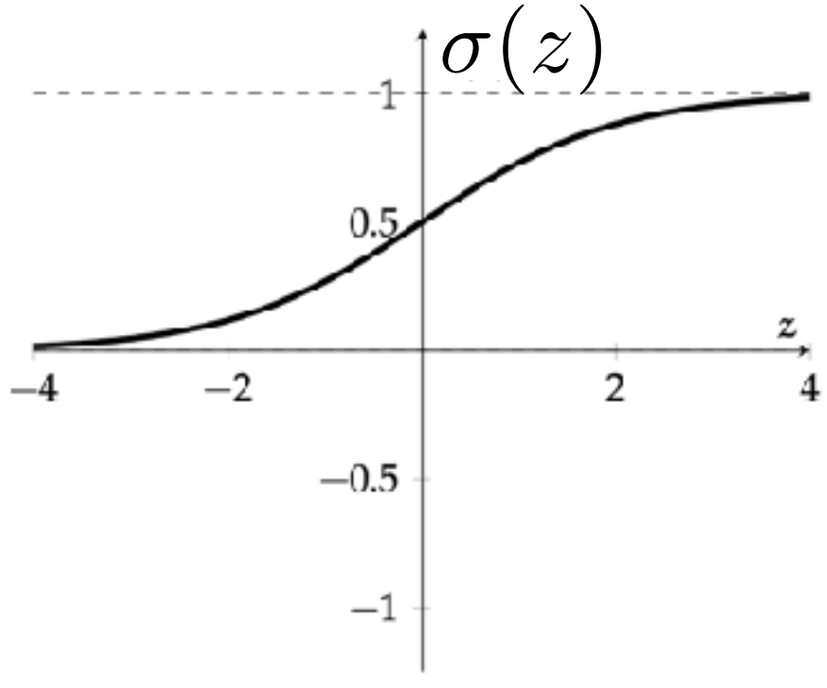
1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

- What if I want to do regression? $f^{(2)}(z) = z$
- What if I want to use NLL loss? $f^{(2)}(z) = \sigma(z)$
- Need non-zero derivatives for (S)GD: Above & $f^{(1)}(z) =$

Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

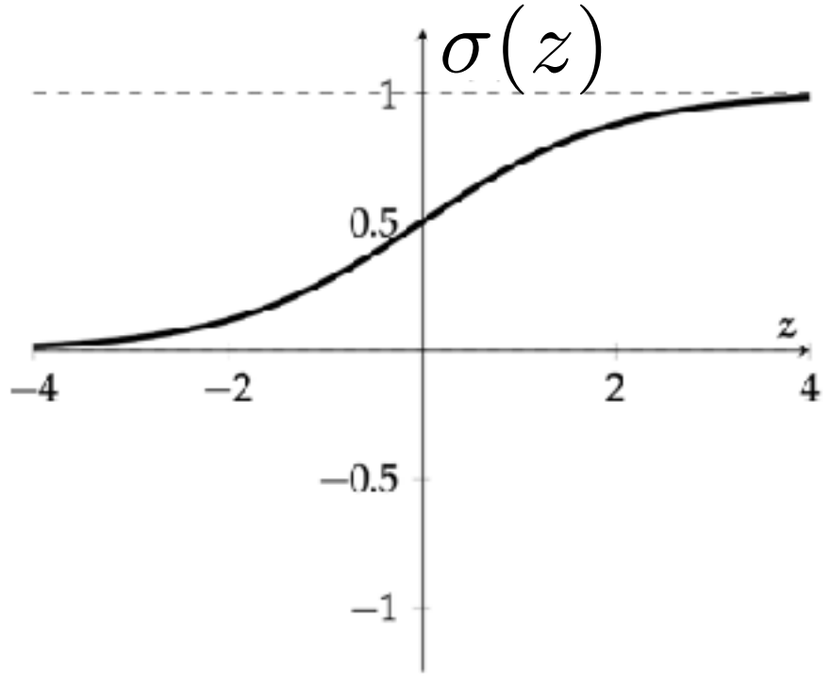
- What if I want to do regression? $f^{(2)}(z) = z$
- What if I want to use NLL loss? $f^{(2)}(z) = \sigma(z)$
- Need non-zero derivatives for (S)GD: Above & $f^{(1)}(z) =$

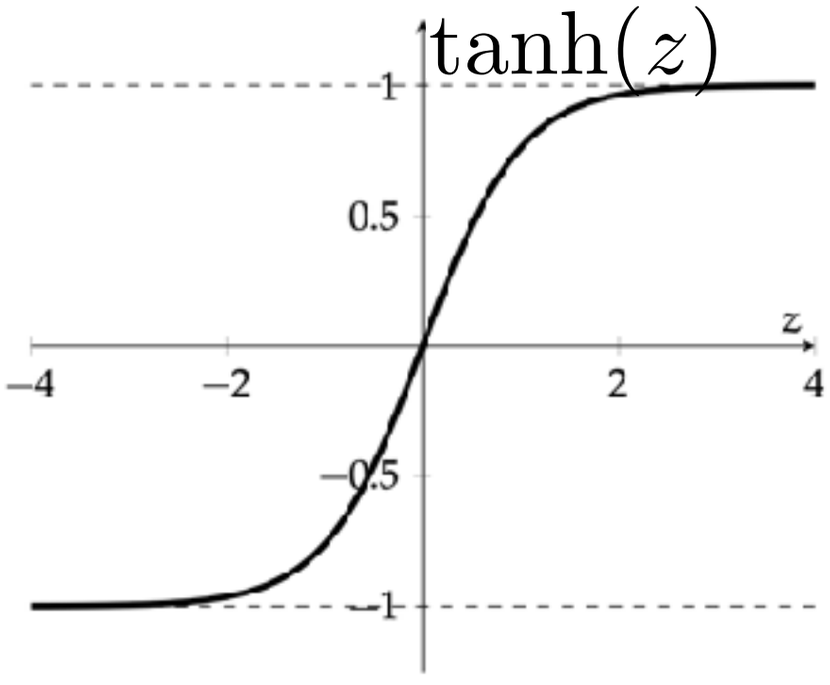


Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

- What if I want to do regression? $f^{(2)}(z) = z$
- What if I want to use NLL loss? $f^{(2)}(z) = \sigma(z)$
- Need non-zero derivatives for (S)GD: Above & $f^{(1)}(z) =$

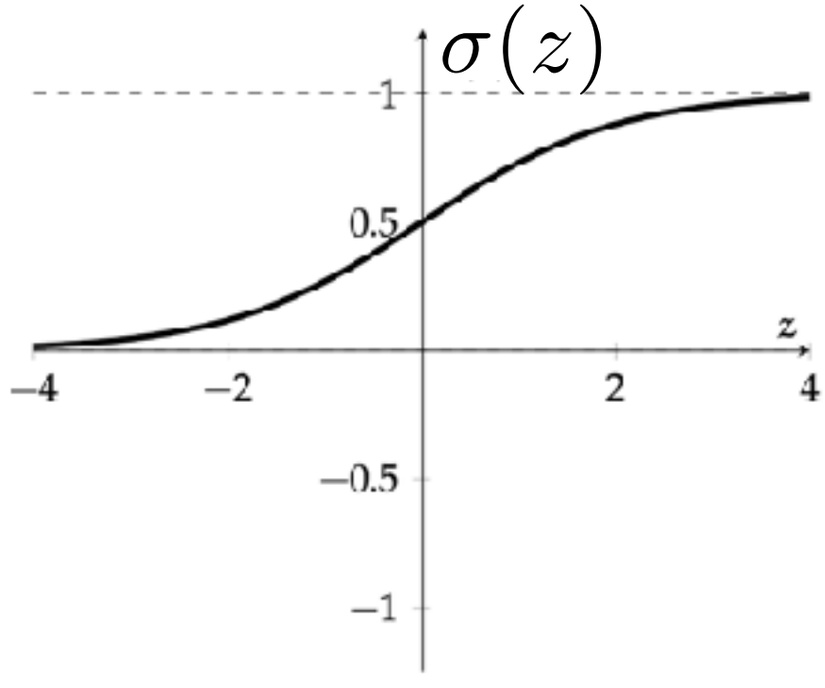


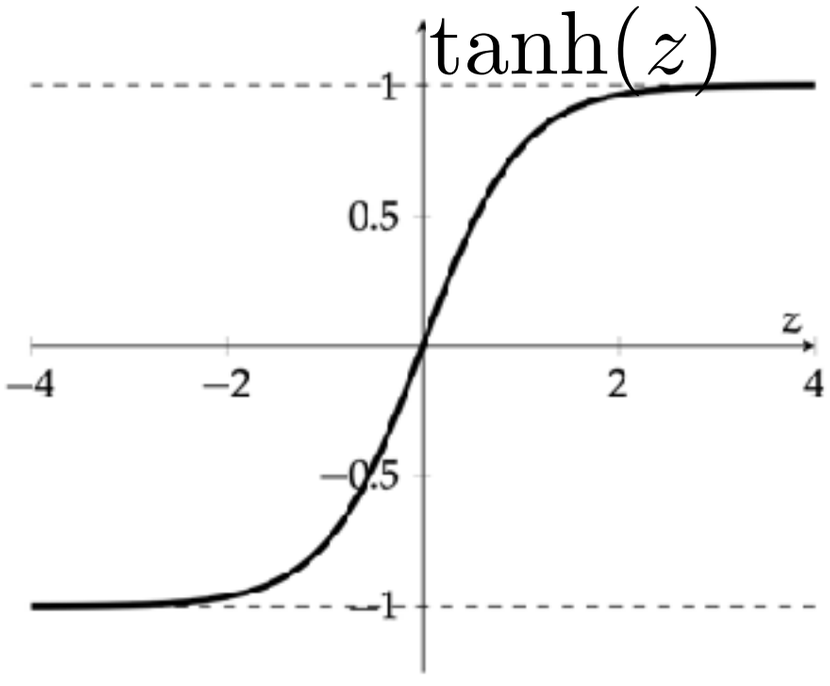


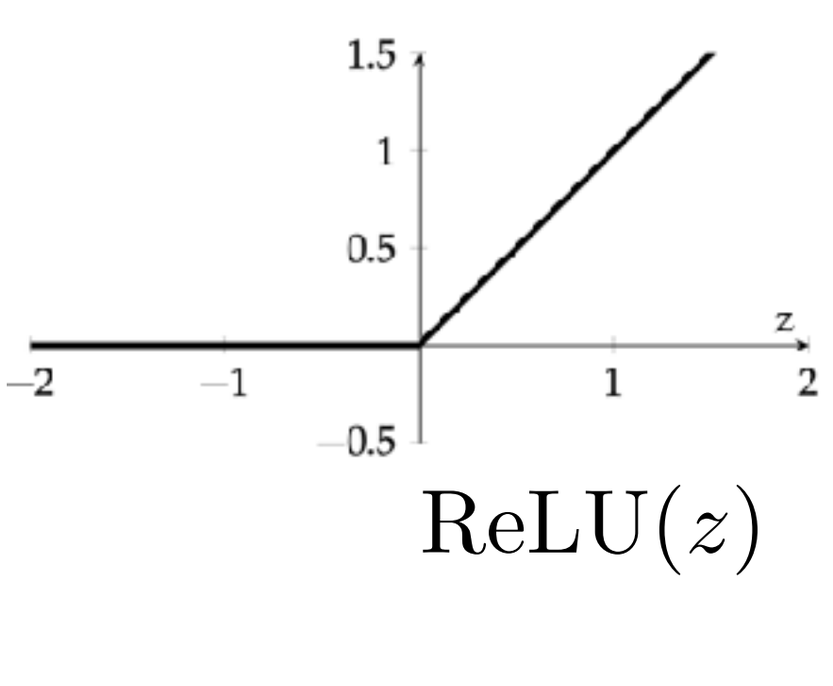
Different activation functions

1. Hypotheses $h(x; W, W_0) = \mathrm{NN}(x; W, W_0)$
   - 1st layer: $A^{(1)} = f^{(1)}(W^{(1)\top} x + W_0^{(1)})$
   - 2nd layer: $A^{(2)} = f^{(2)}(W^{(2)\top} A^{(1)} + W_0^{(2)})$

- What if I want to do regression? $f^{(2)}(z) = z$
- What if I want to use NLL loss? $f^{(2)}(z) = \sigma(z)$
- Need non-zero derivatives for (S)GD: Above & $f^{(1)}(z) =$







Choices of activation function

Choices of activation function

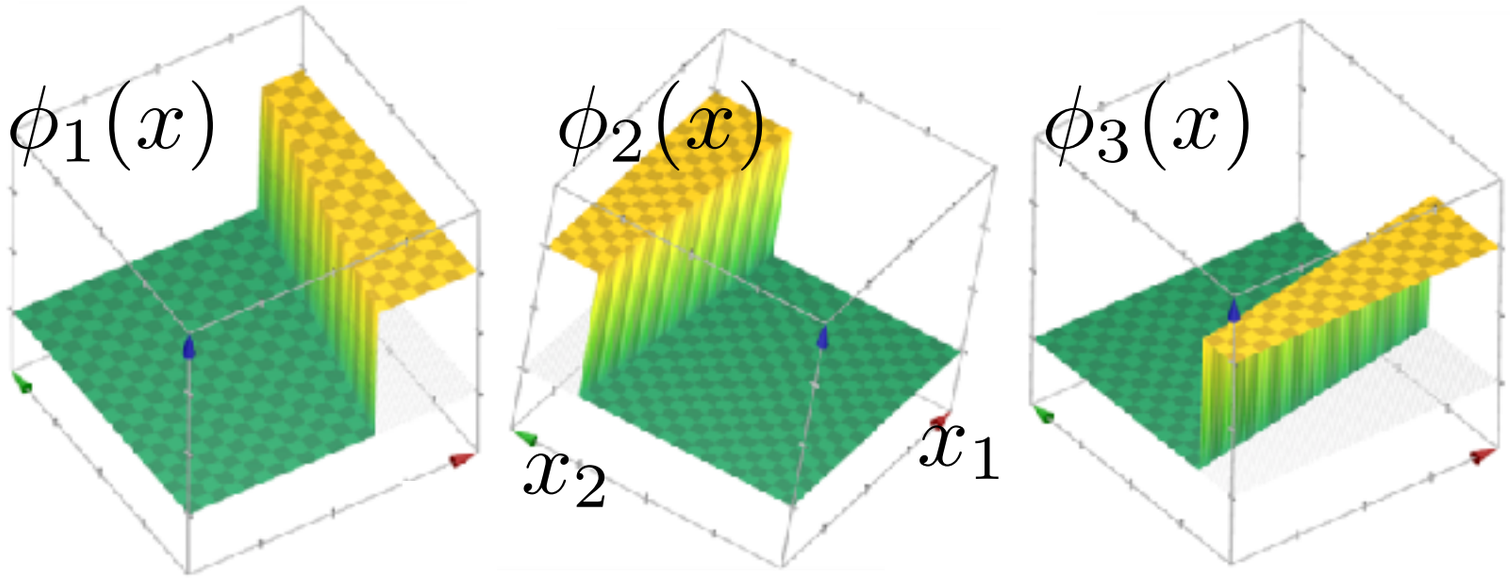
- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$

Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose
    $f^{(1)}(z) = \mathbf{1}\{z \geq 0\}$

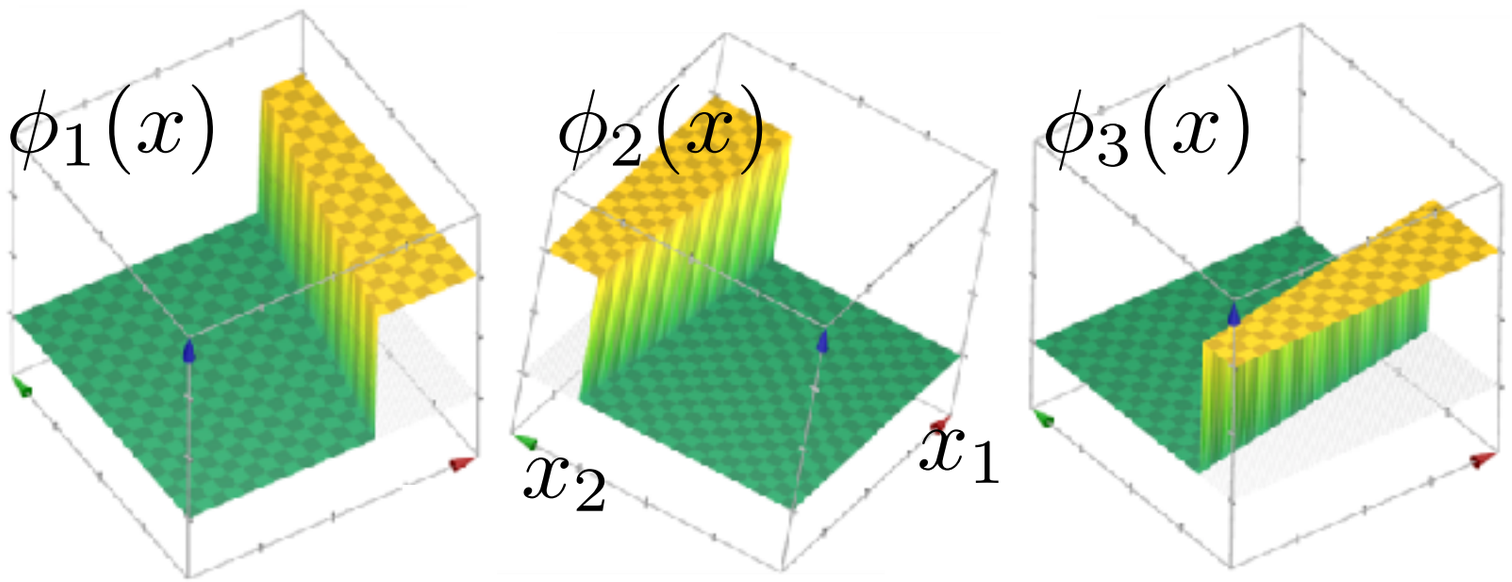
Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose
    $f^{(1)}(z) = \mathbf{1}\{z \geq 0\}$



Choices of activation function

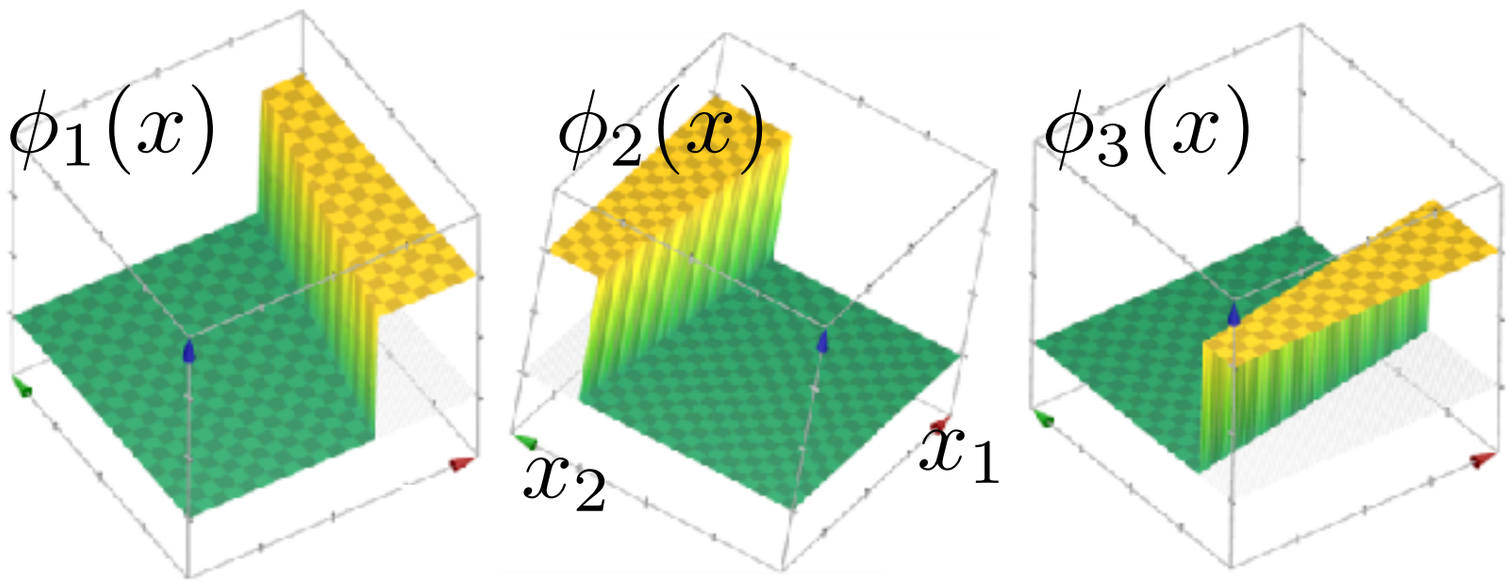
- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \mathbf{1}\{z \geq 0\}$



- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$

Choices of activation function

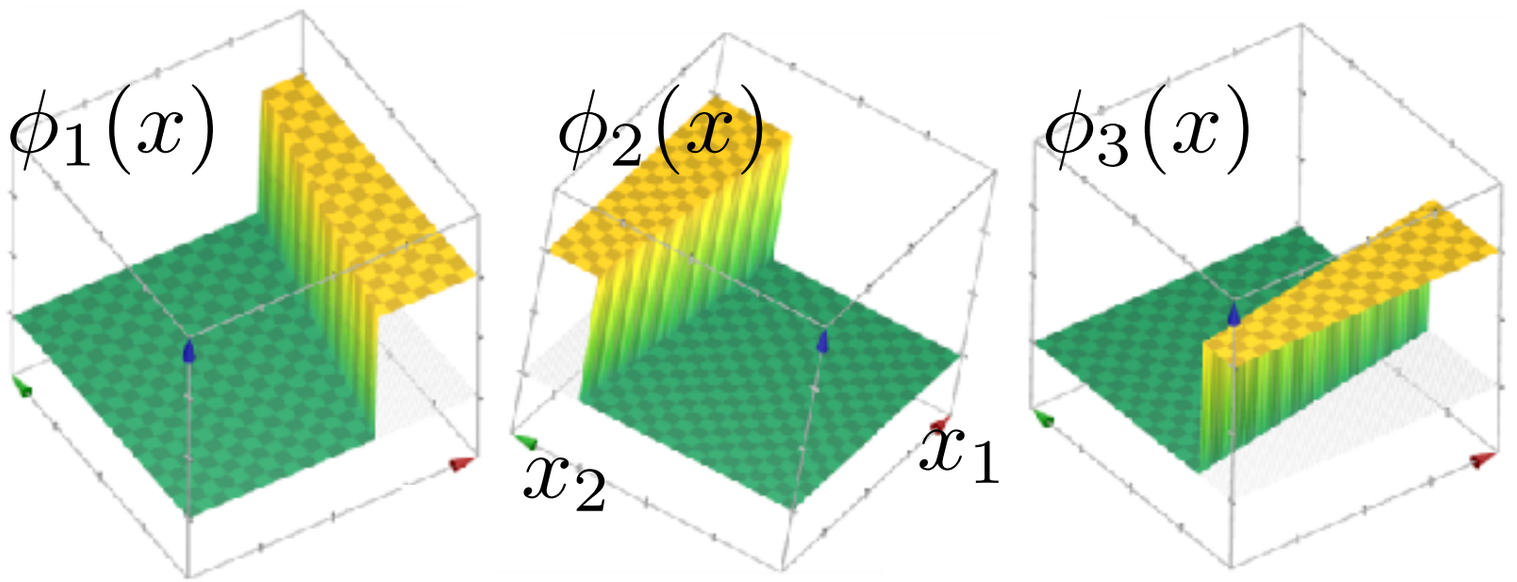
- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose
    $f^{(1)}(z) = \mathbf{1}\{z \geq 0\}$



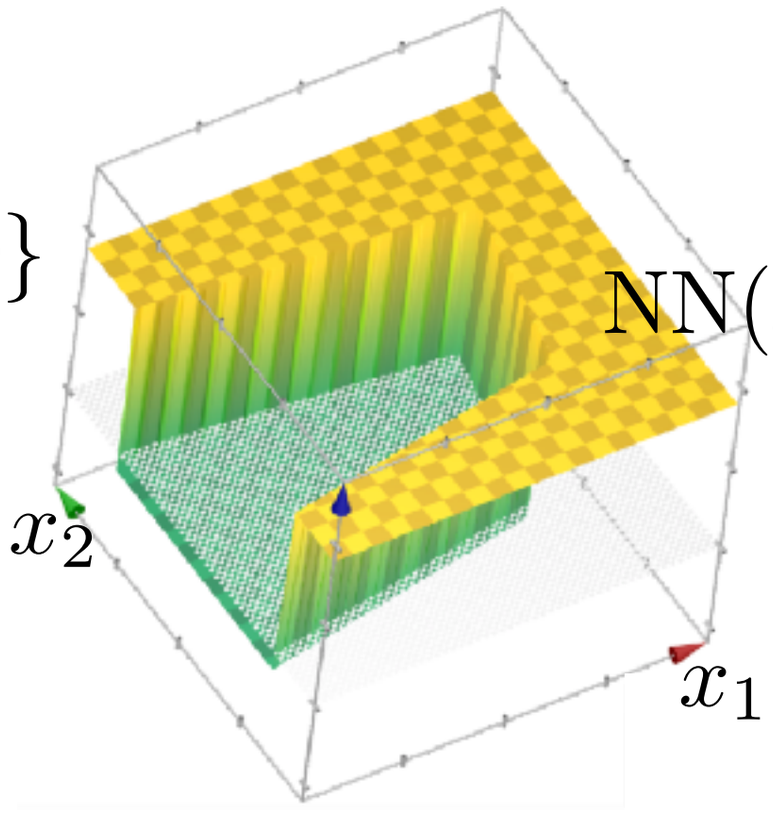
- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose
    $f^{(2)}(z) = \mathbf{1}\{z \geq 0\}$

Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose
    $f^{(1)}(z) = \mathbf{1}\{z \geq 0\}$



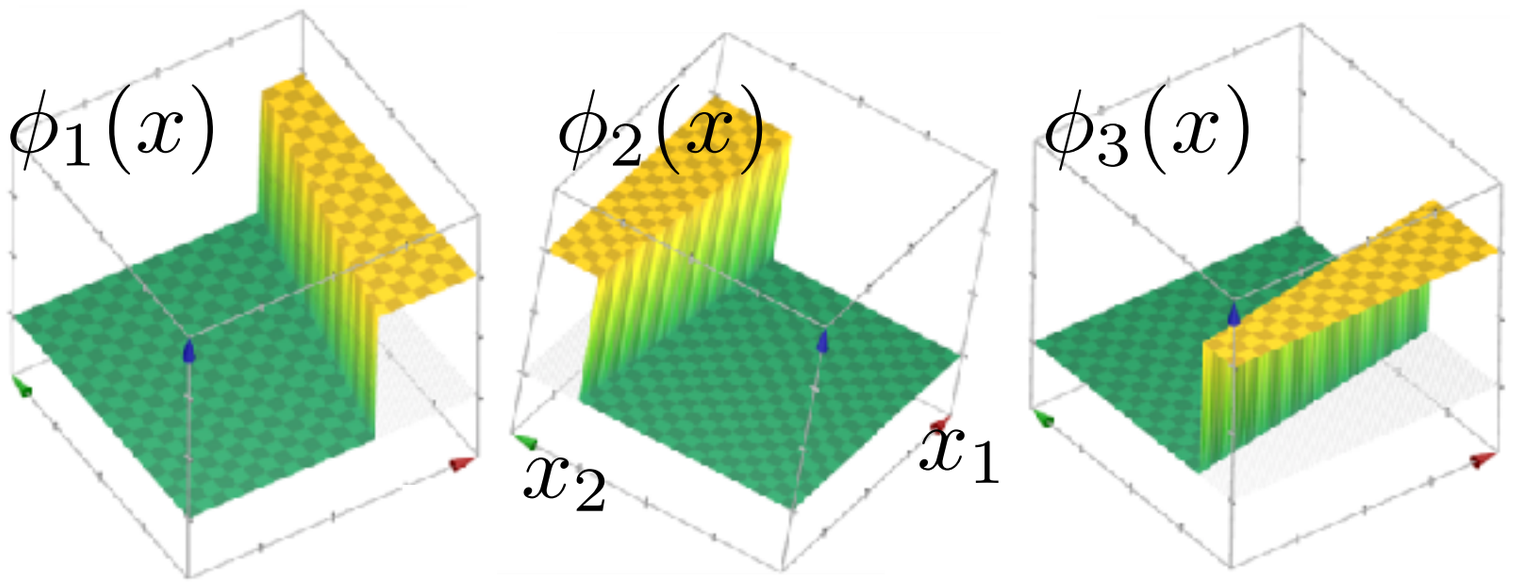
- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose
    $f^{(2)}(z) = \mathbf{1}\{z \geq 0\}$



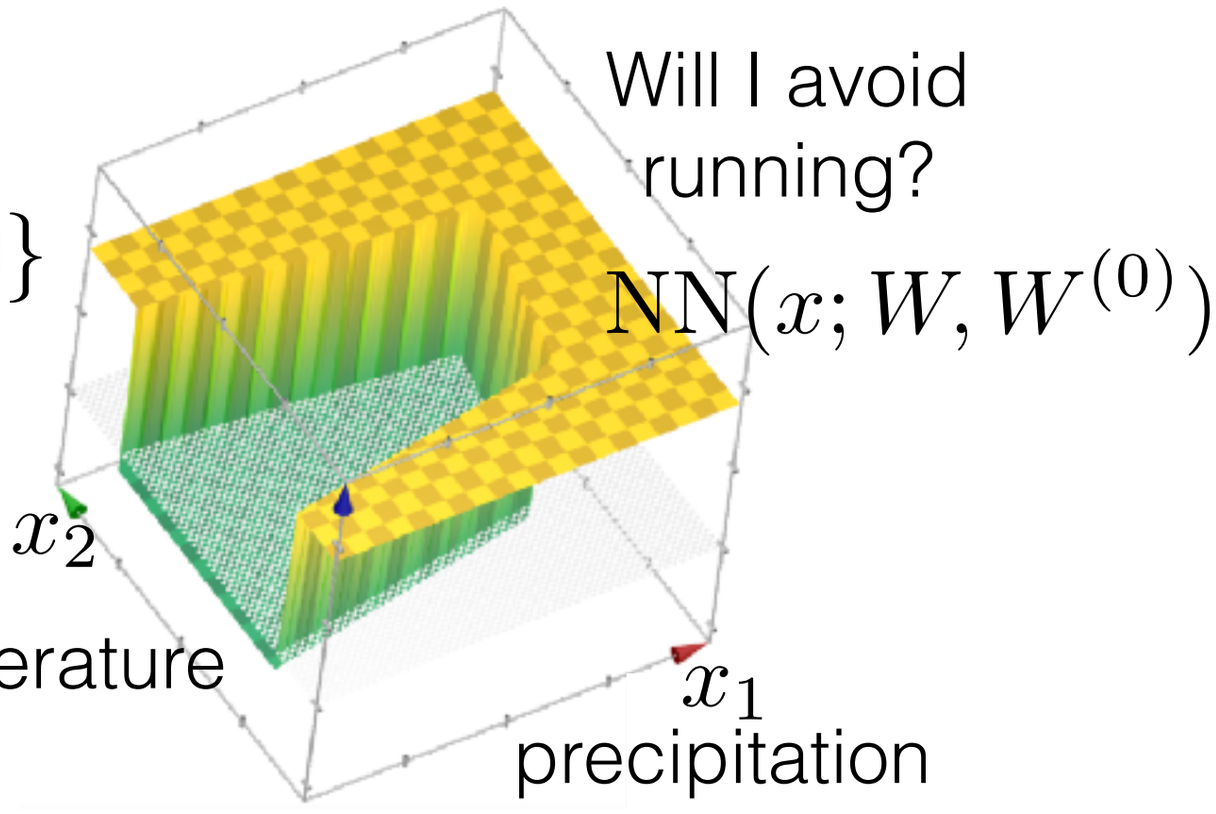
$\mathrm{NN}(x; W, W^{(0)})$

Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \mathbf{1}\{z \geq 0\}$



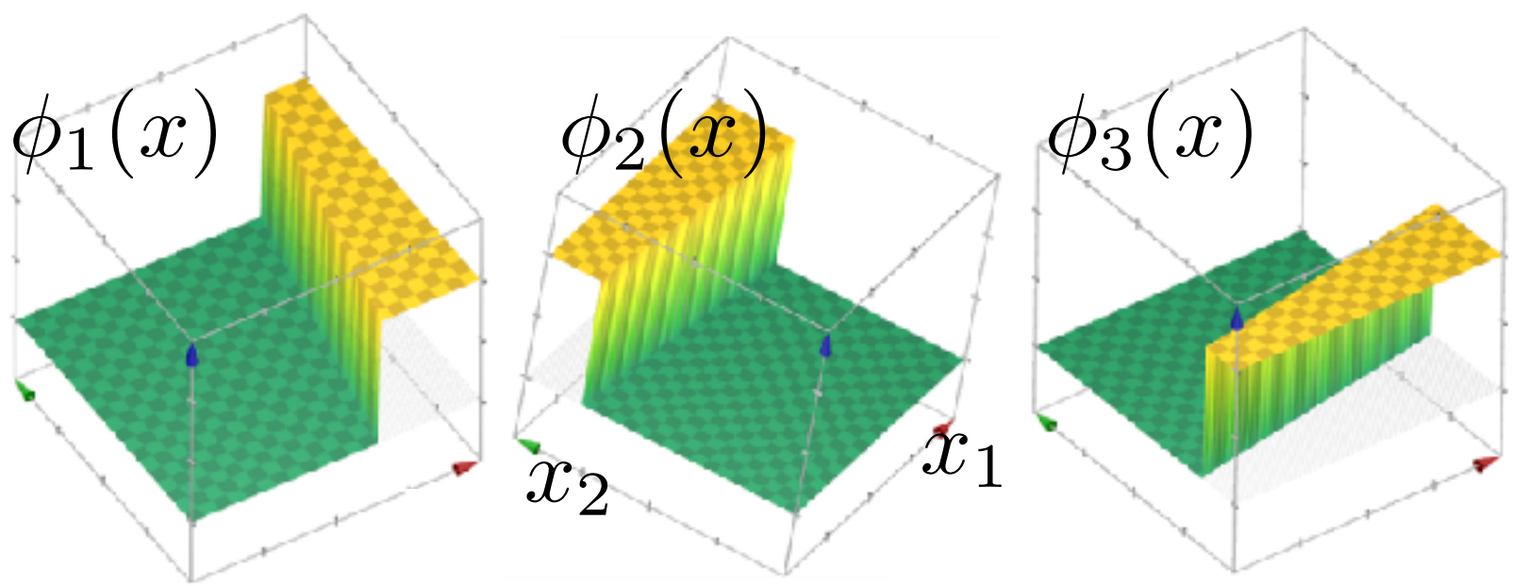
- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = \mathbf{1}\{z \geq 0\}$



Choices of activation function

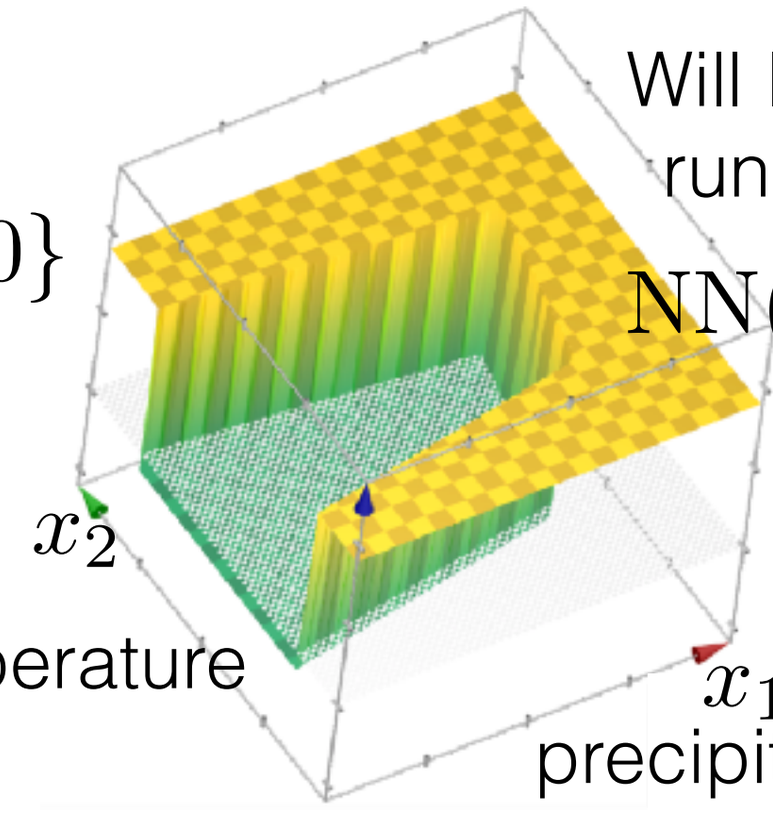
- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose

$$f^{(1)}(z) = \mathbf{1}\{z \geq 0\}$$



- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose

$$f^{(2)}(z) = \mathbf{1}\{z \geq 0\}$$



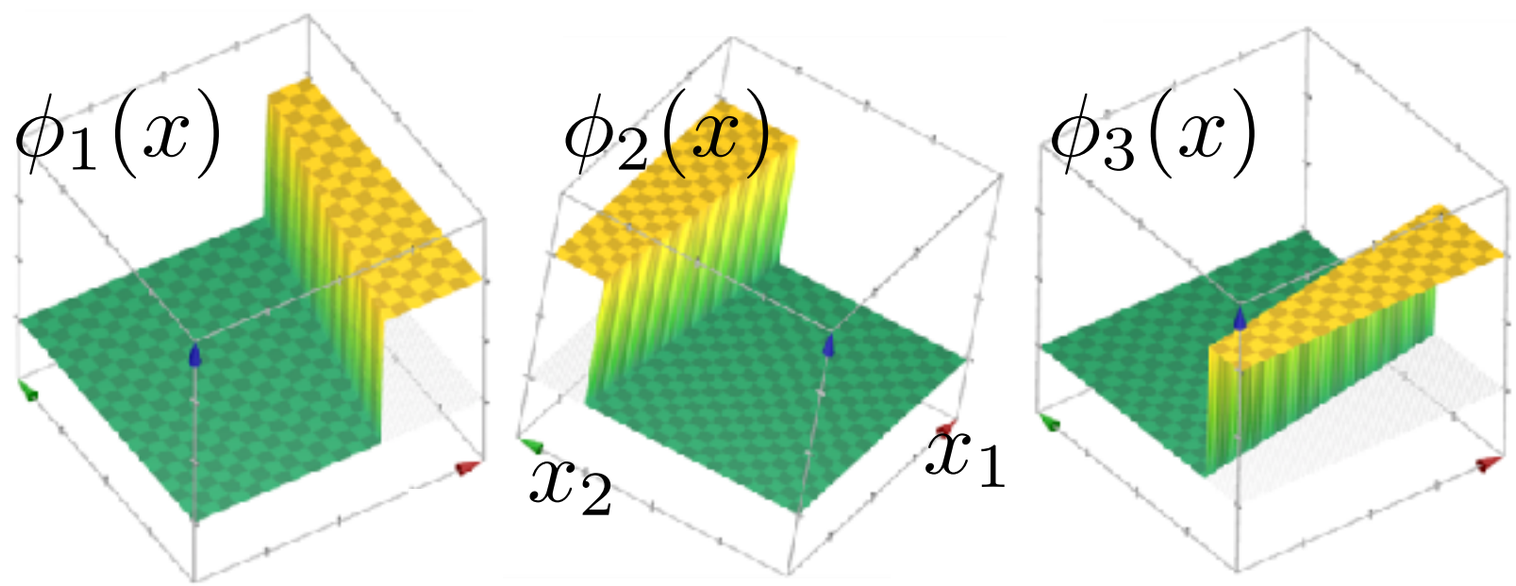
Will I avoid running?

$$\mathrm{NN}(x; W, W^{(0)})$$

Choices of activation function

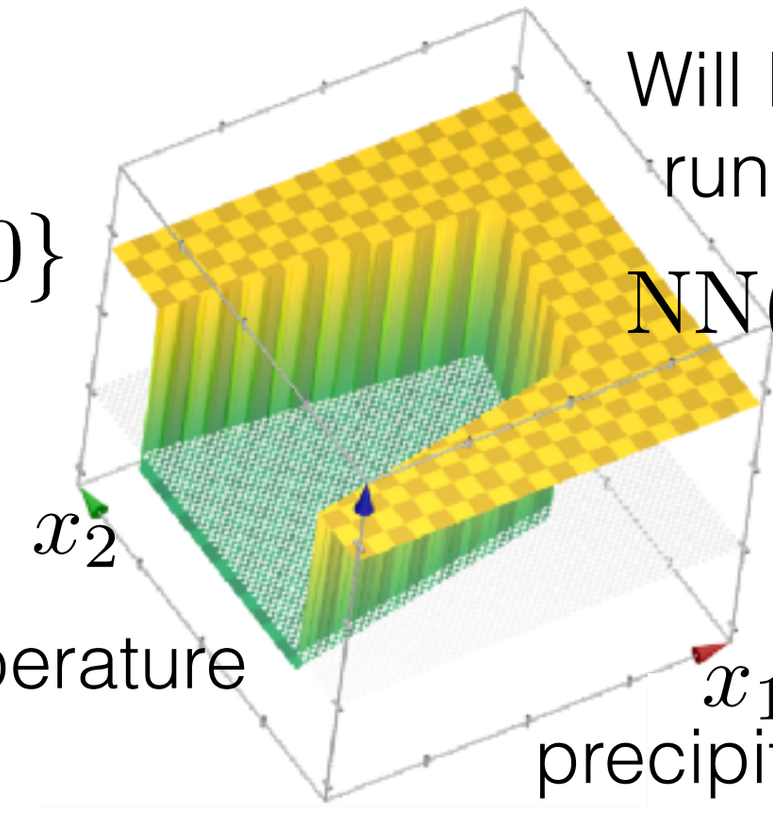
- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose

$$f^{(1)}(z) = \sigma(z)$$



- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose

$$f^{(2)}(z) = \mathbf{1}\{z \geq 0\}$$

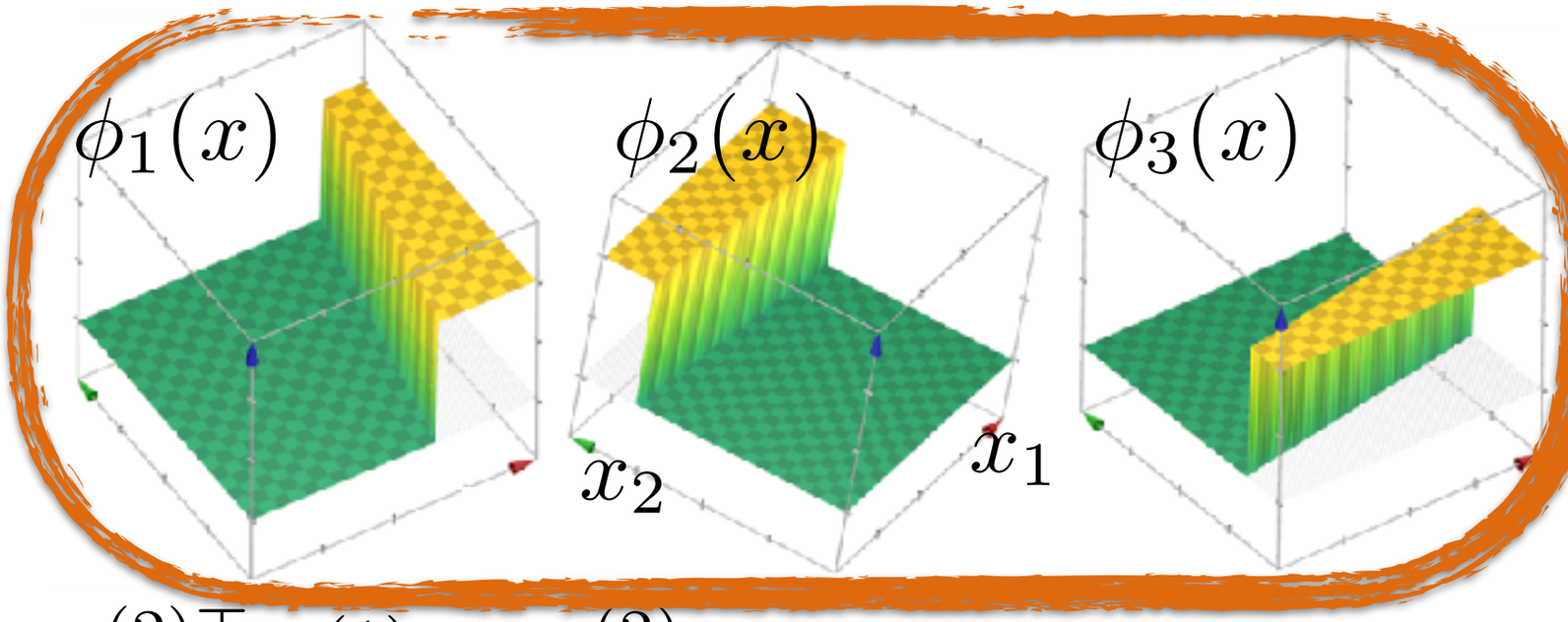


Will I avoid running?

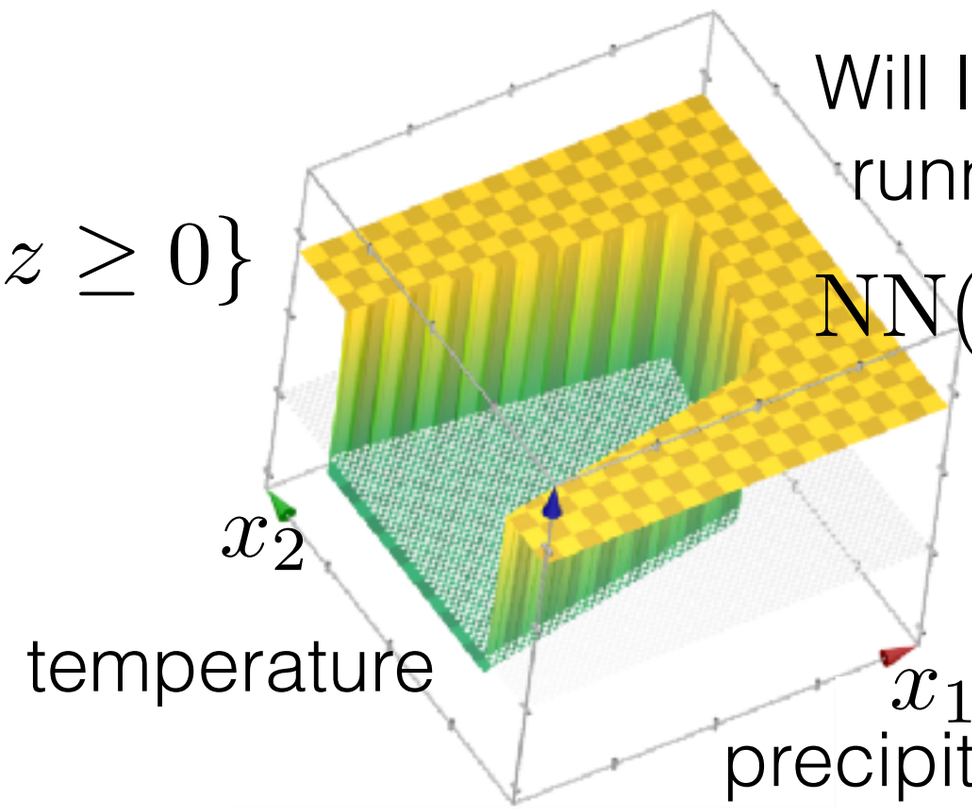
$$\mathrm{NN}(x; W, W^{(0)})$$

Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$



- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = \mathbf{1}\{z \geq 0\}$

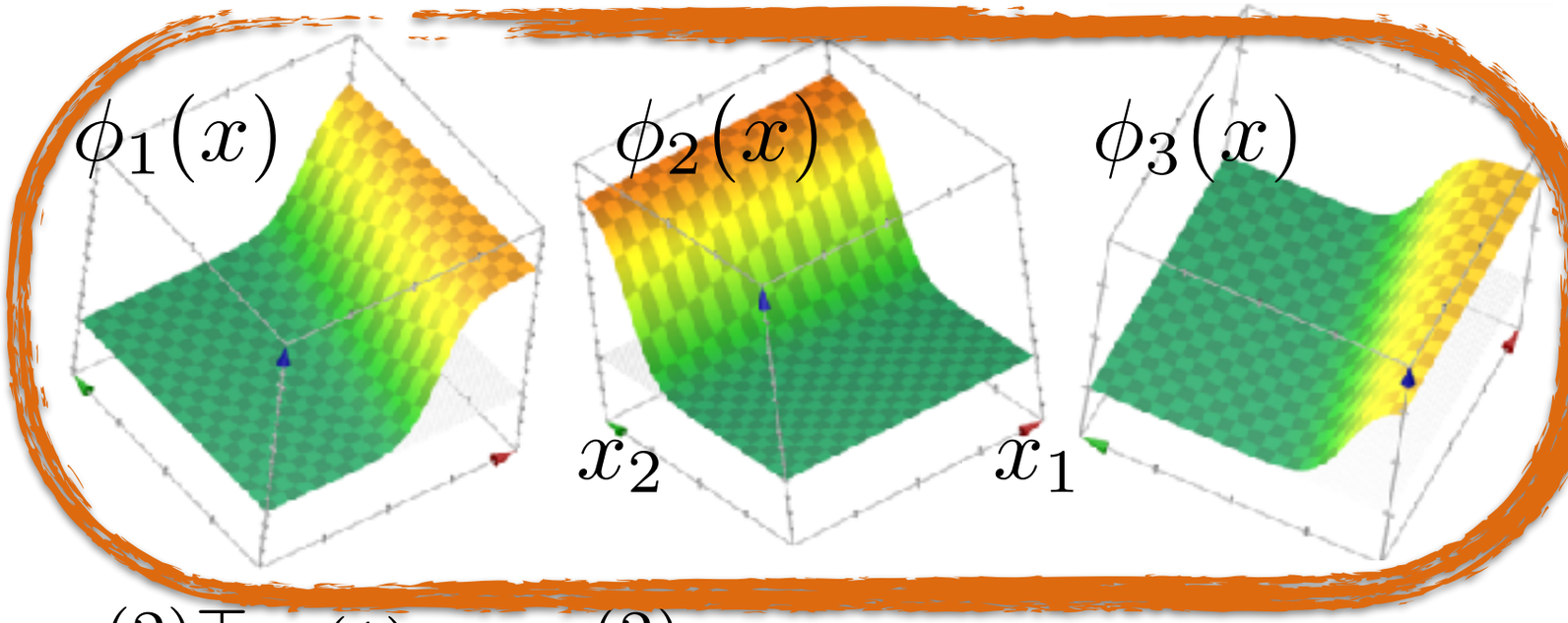


Will I avoid running?

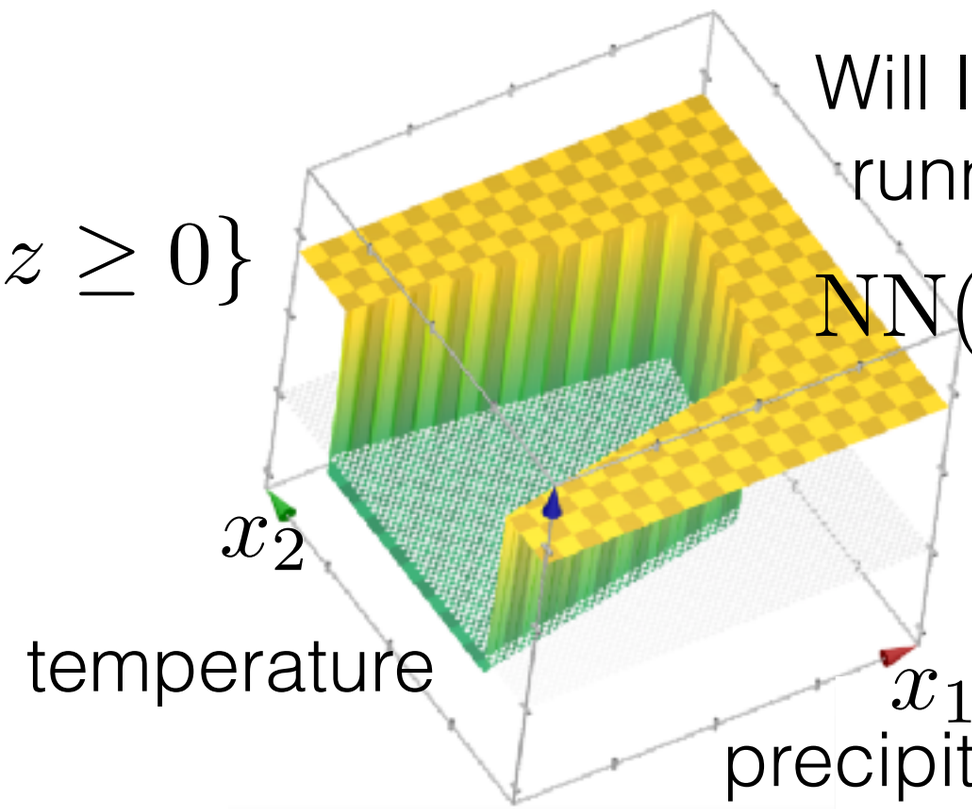
$\mathrm{NN}(x; W, W^{(0)})$

Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$

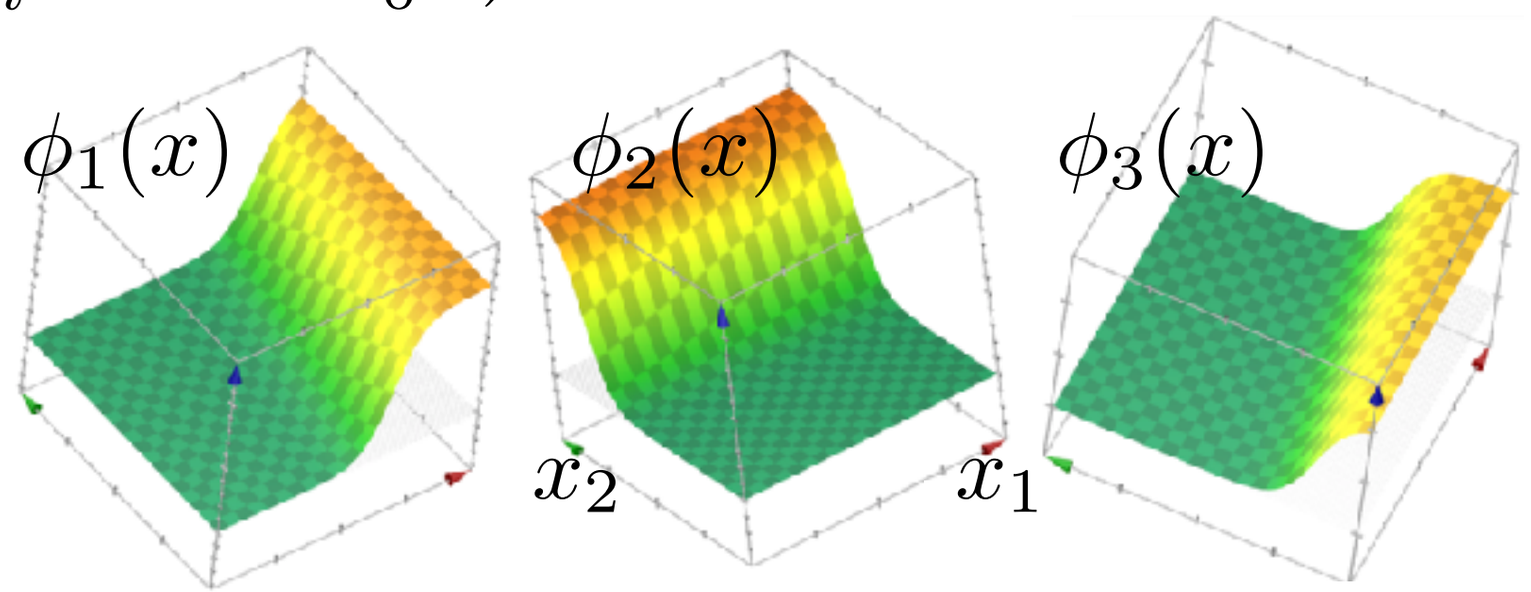


- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = \mathbf{1}\{z \geq 0\}$

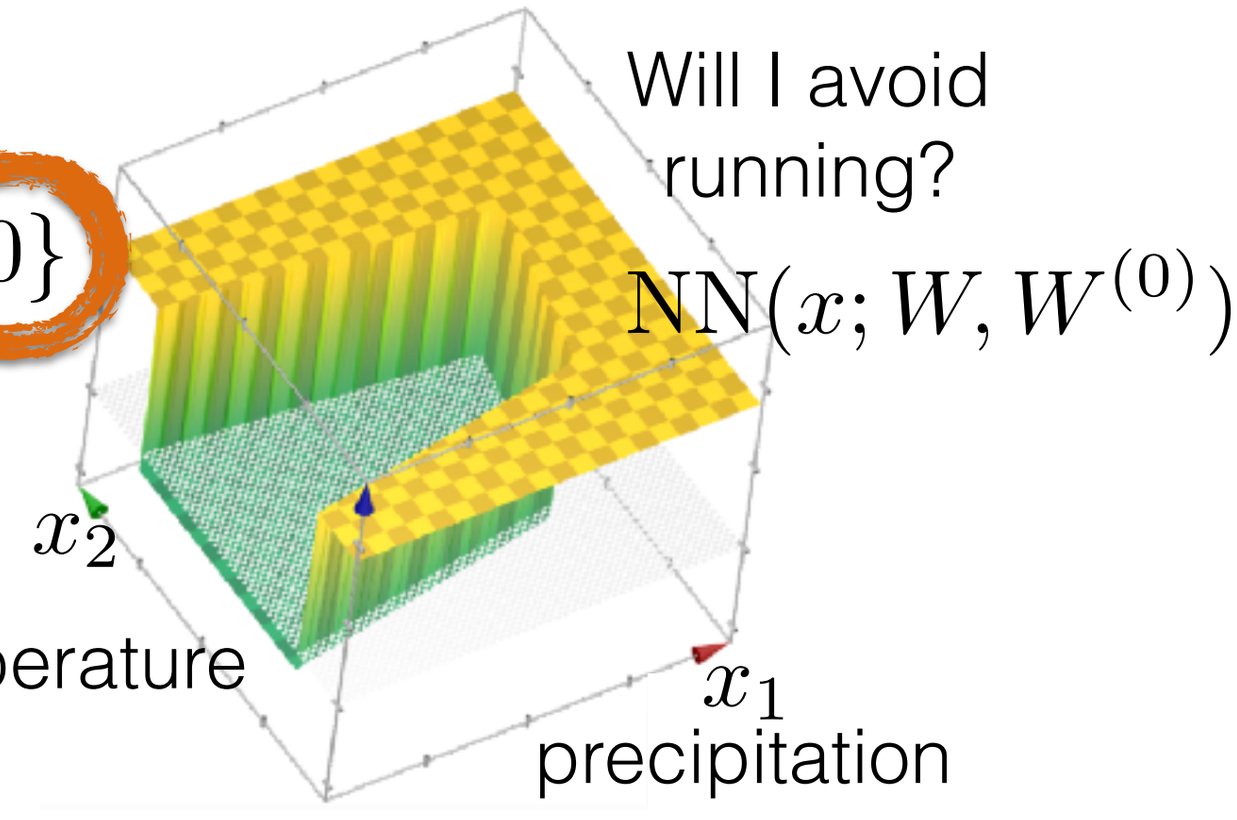


Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$

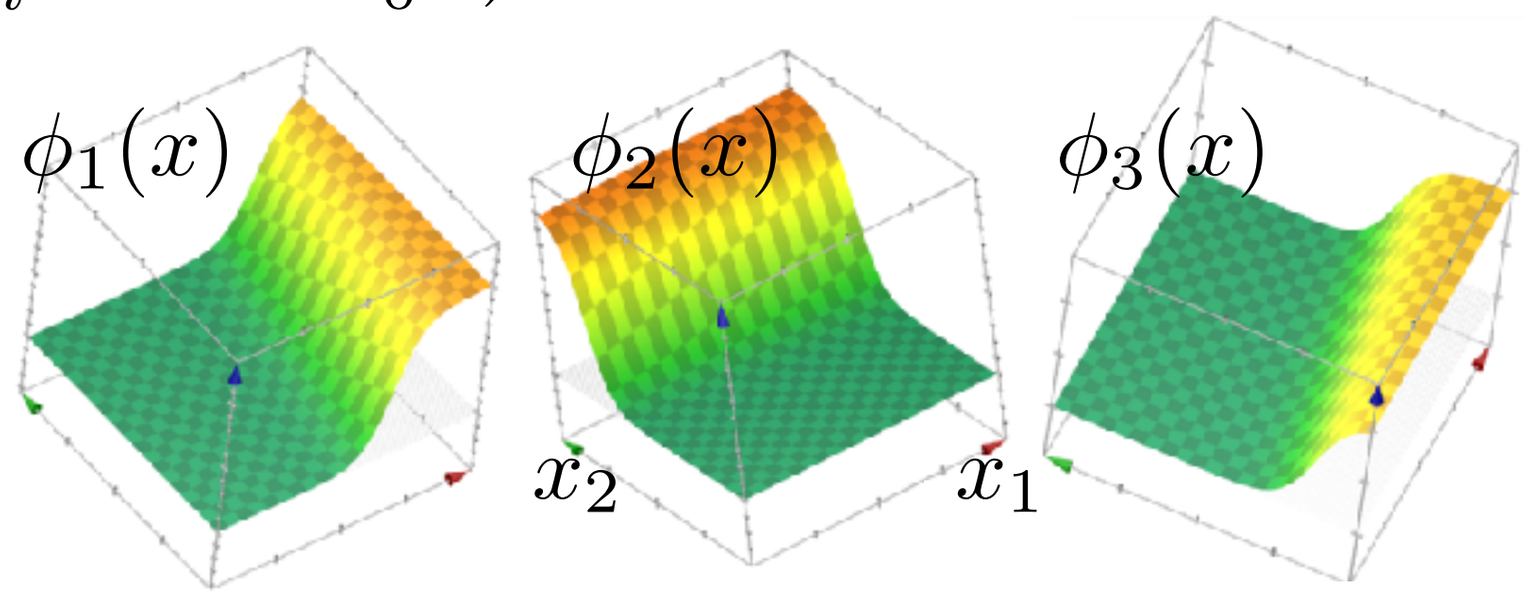


- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = \mathbf{1}\{z \geq 0\}$

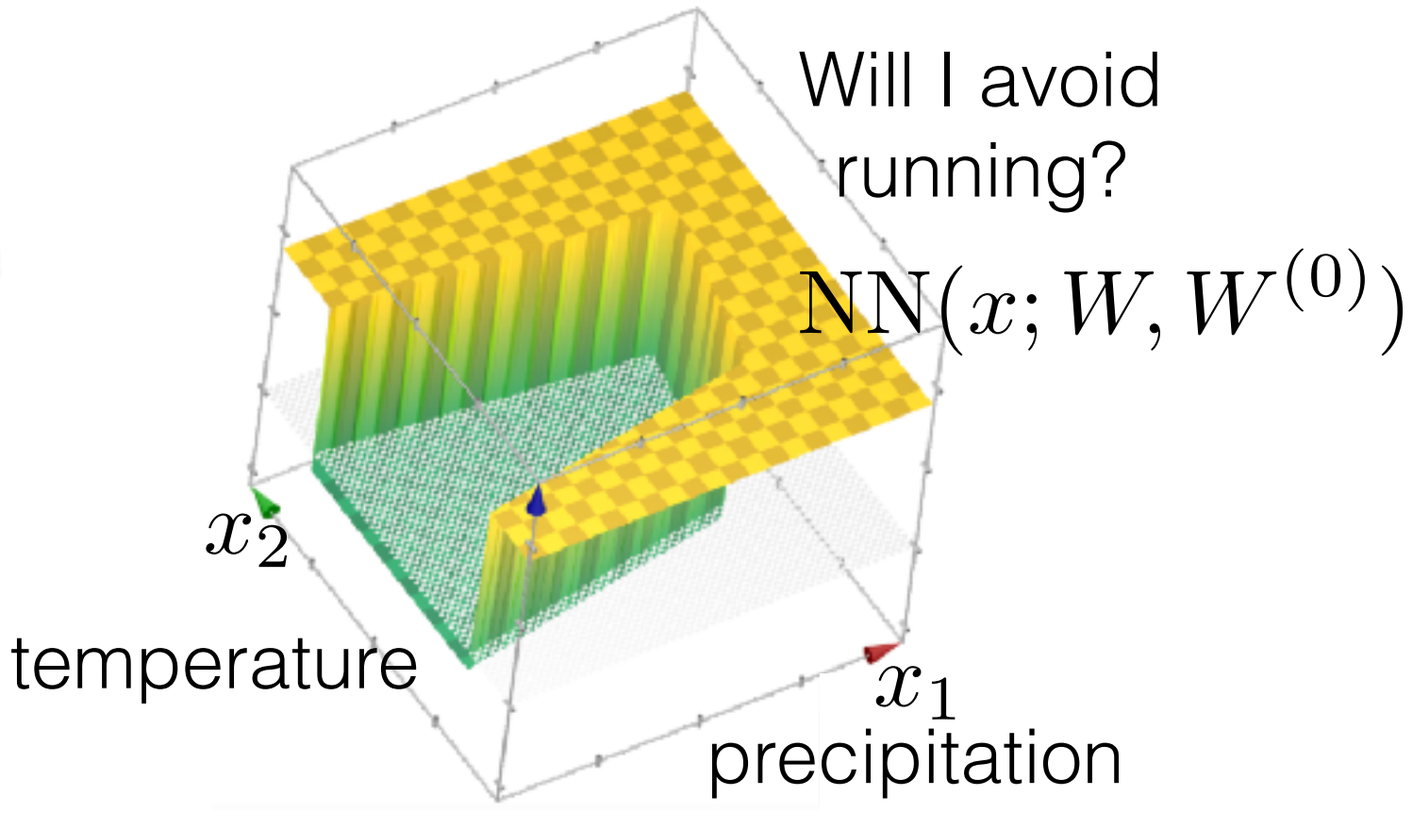


Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$

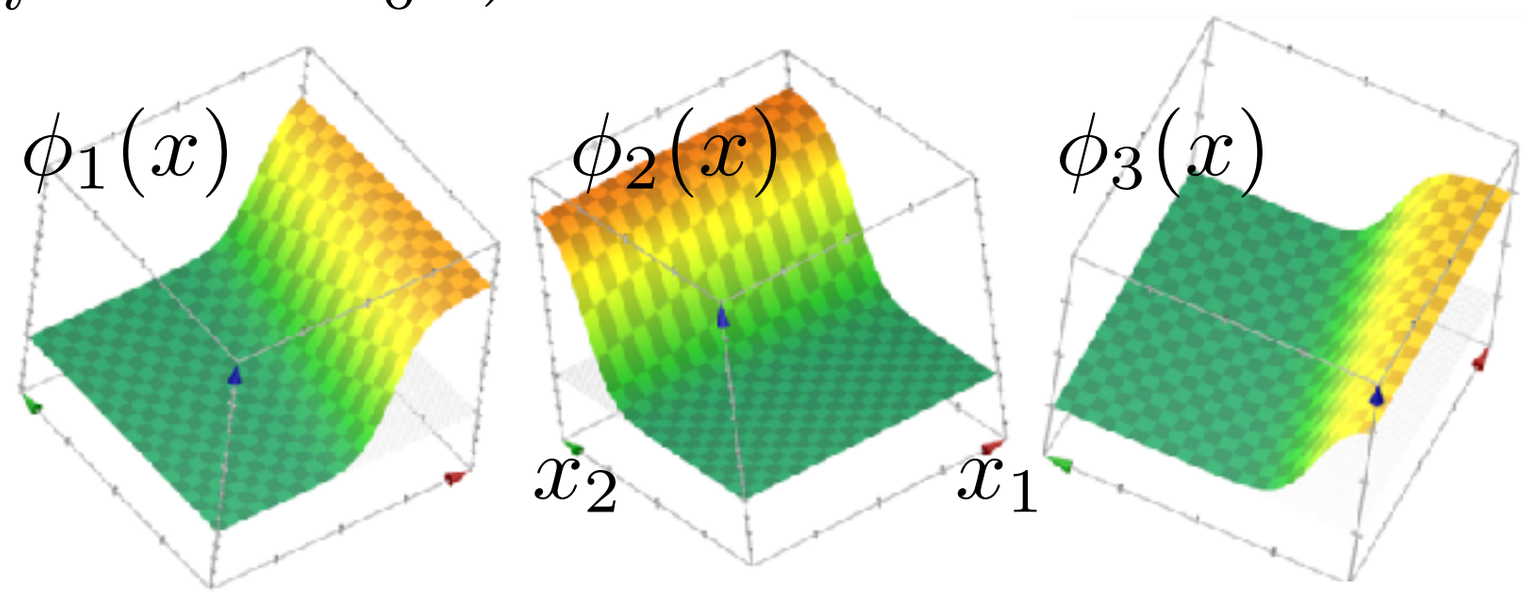


- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = z$

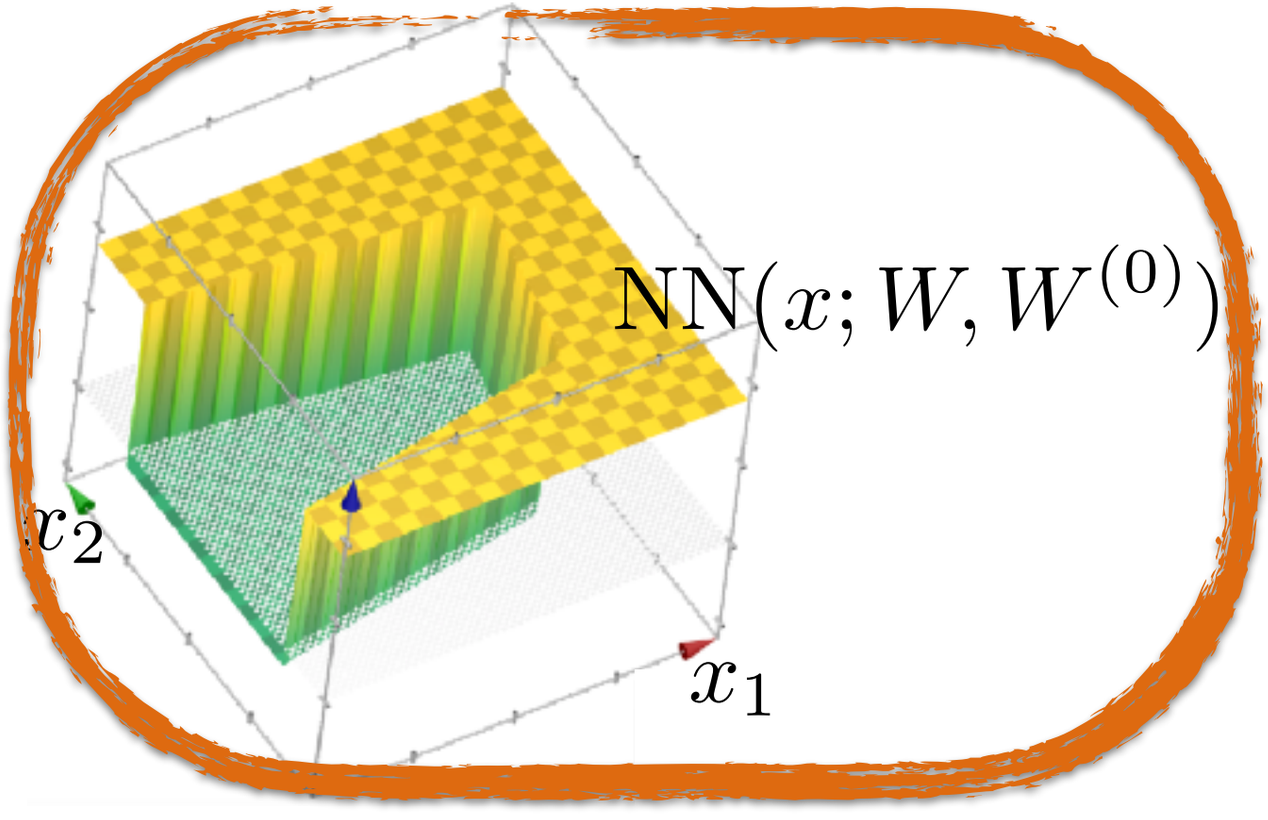


Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$

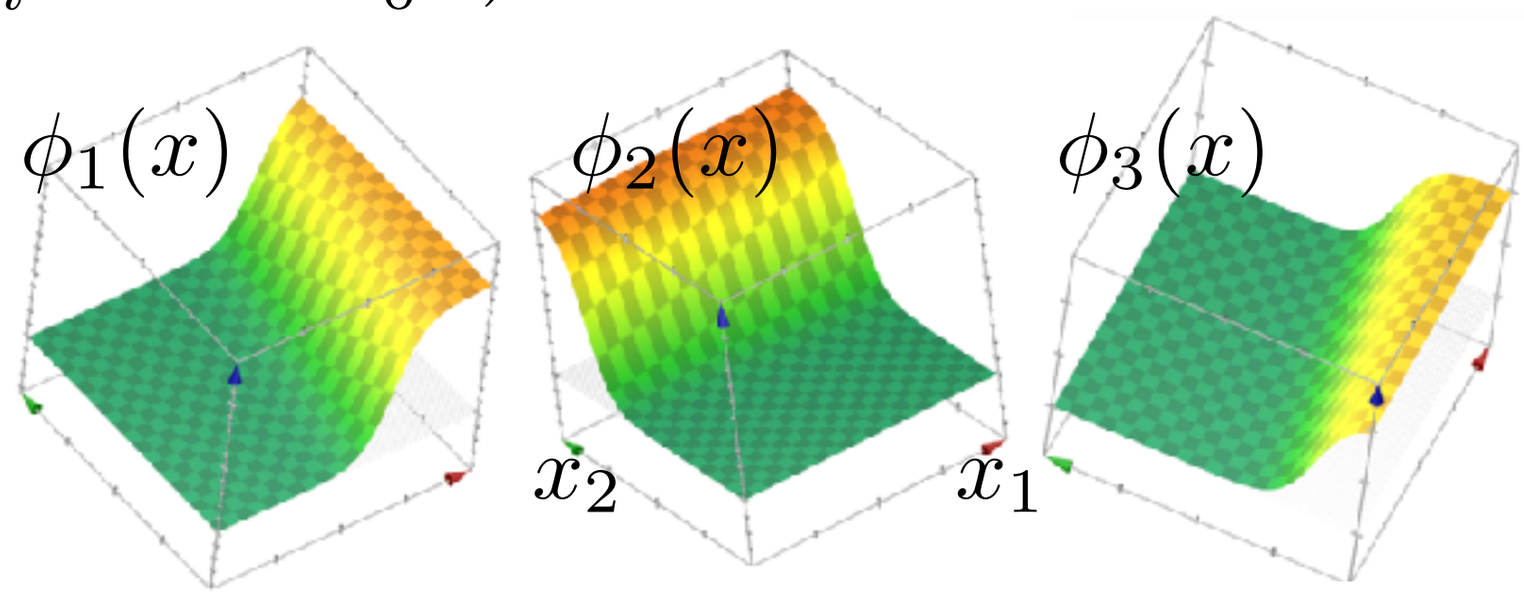


- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = z$

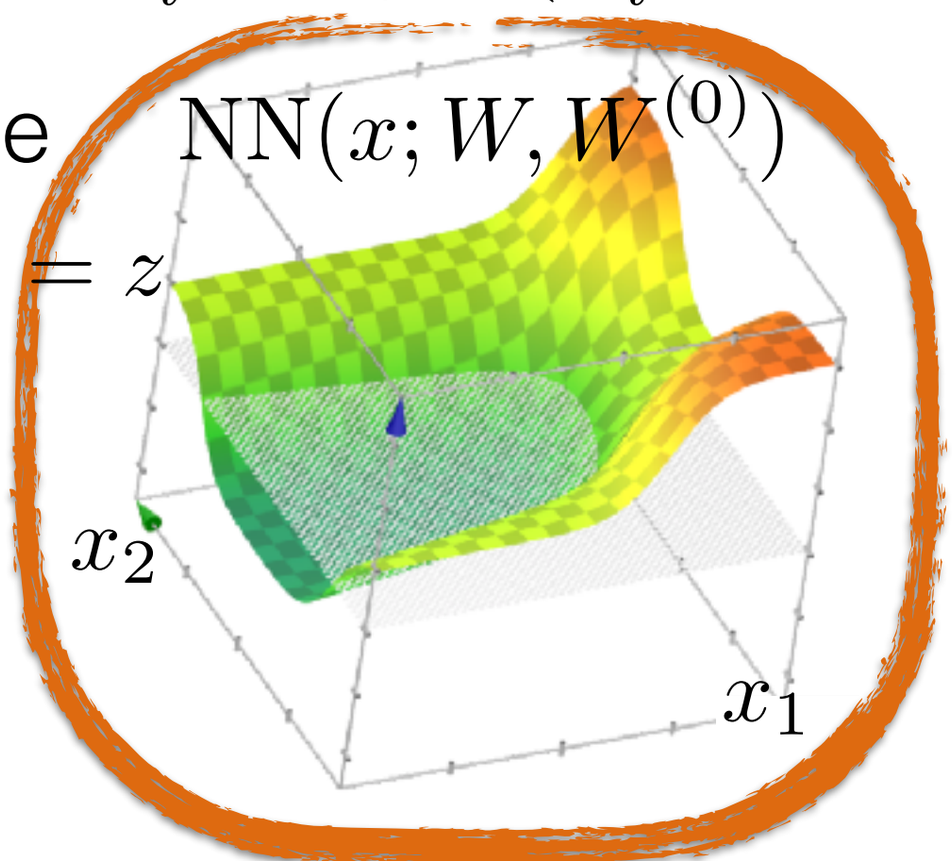


Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$

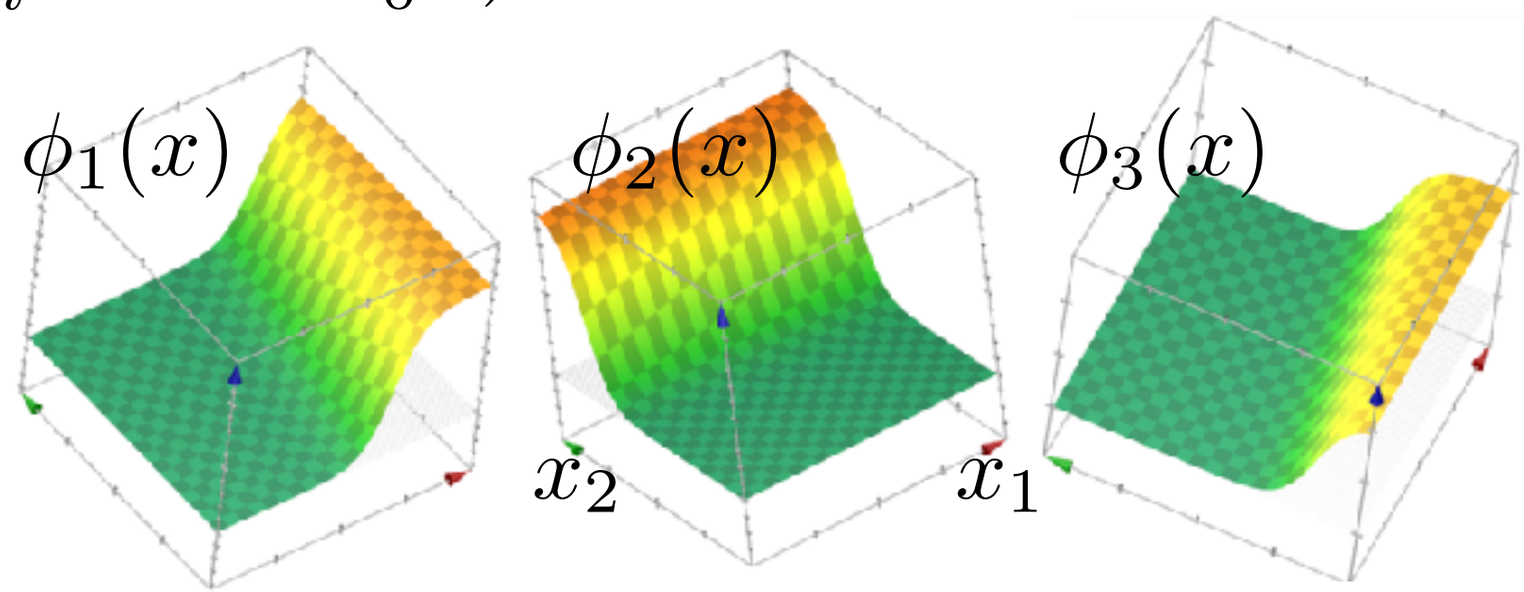


- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = z$

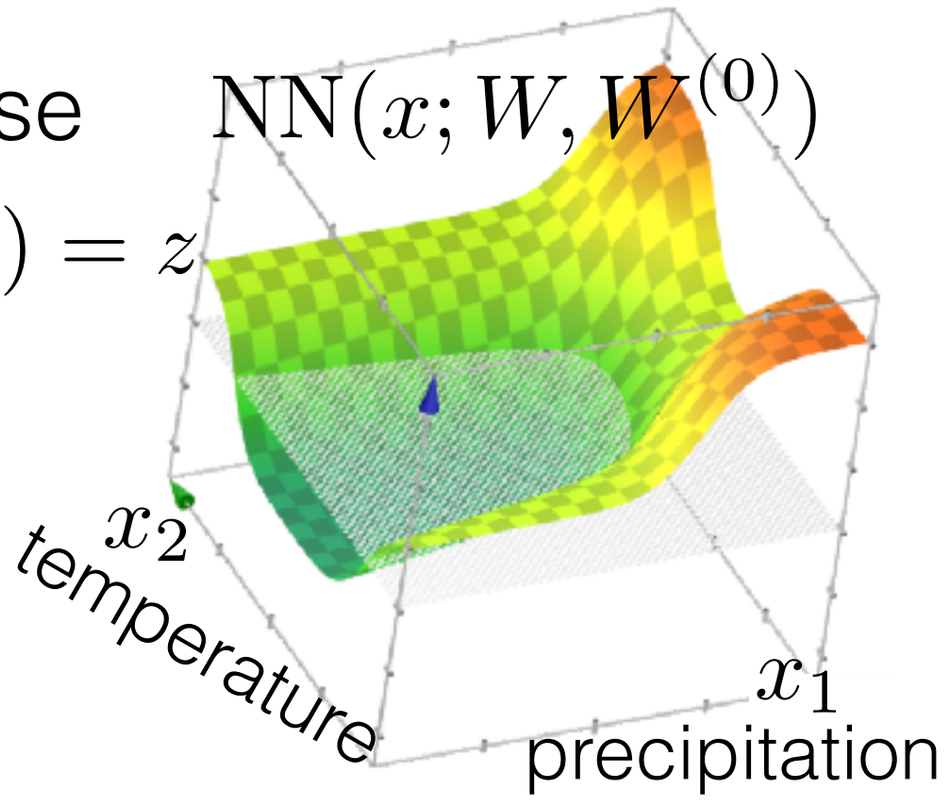


Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$

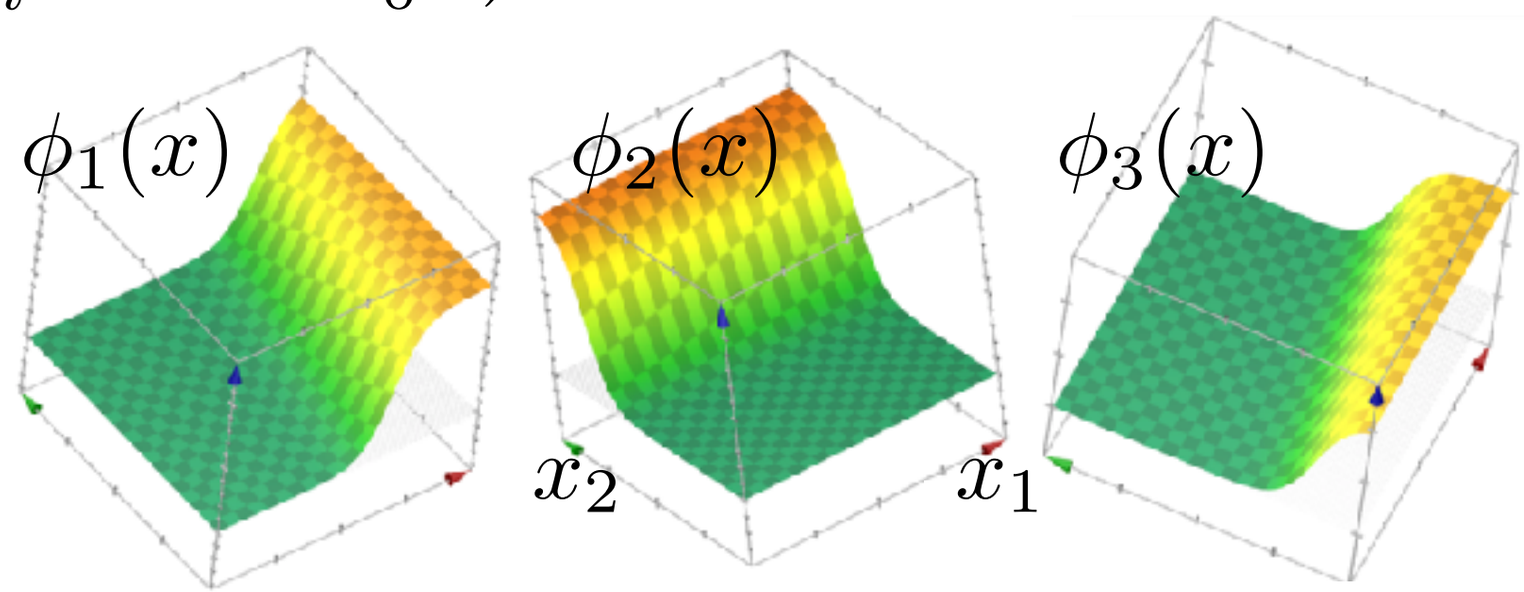


- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = z$



Choices of activation function

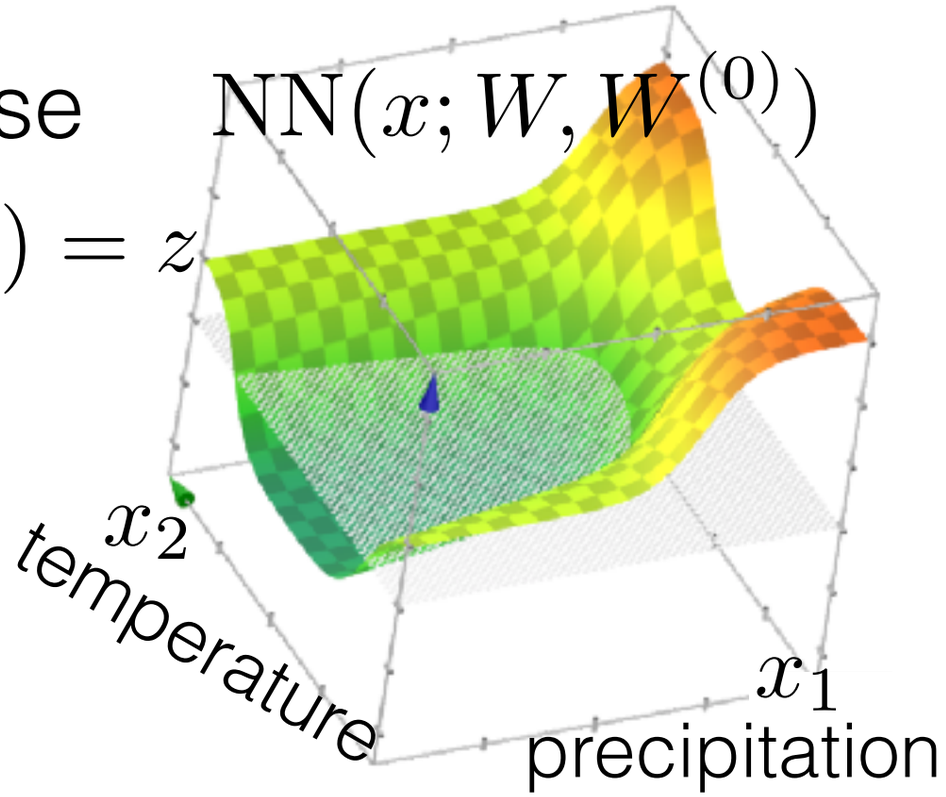
- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$



- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = z$

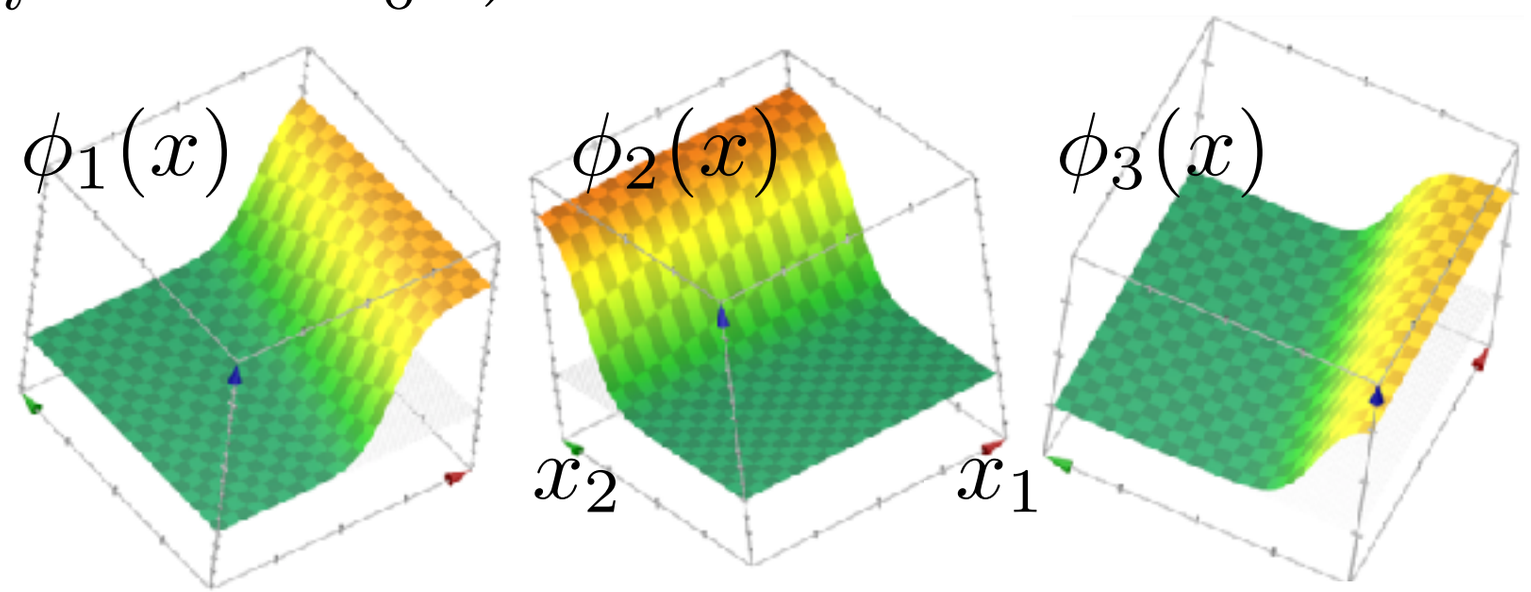
$\mathrm{NN}(x; W, W^{(0)})$

How much will I dislike running today?

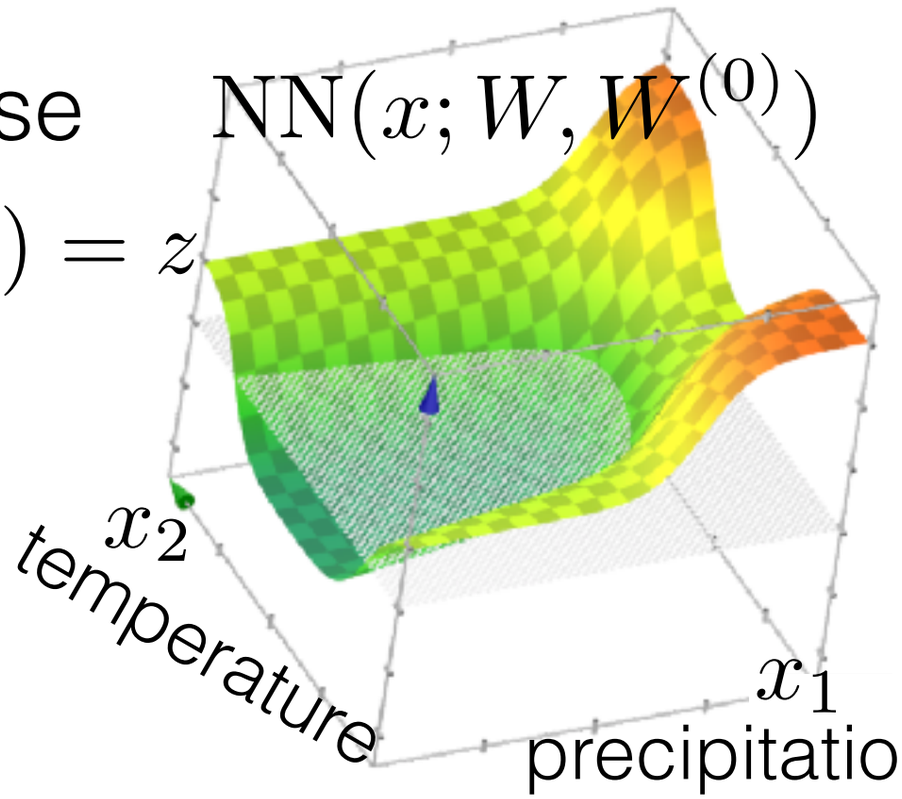


Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$

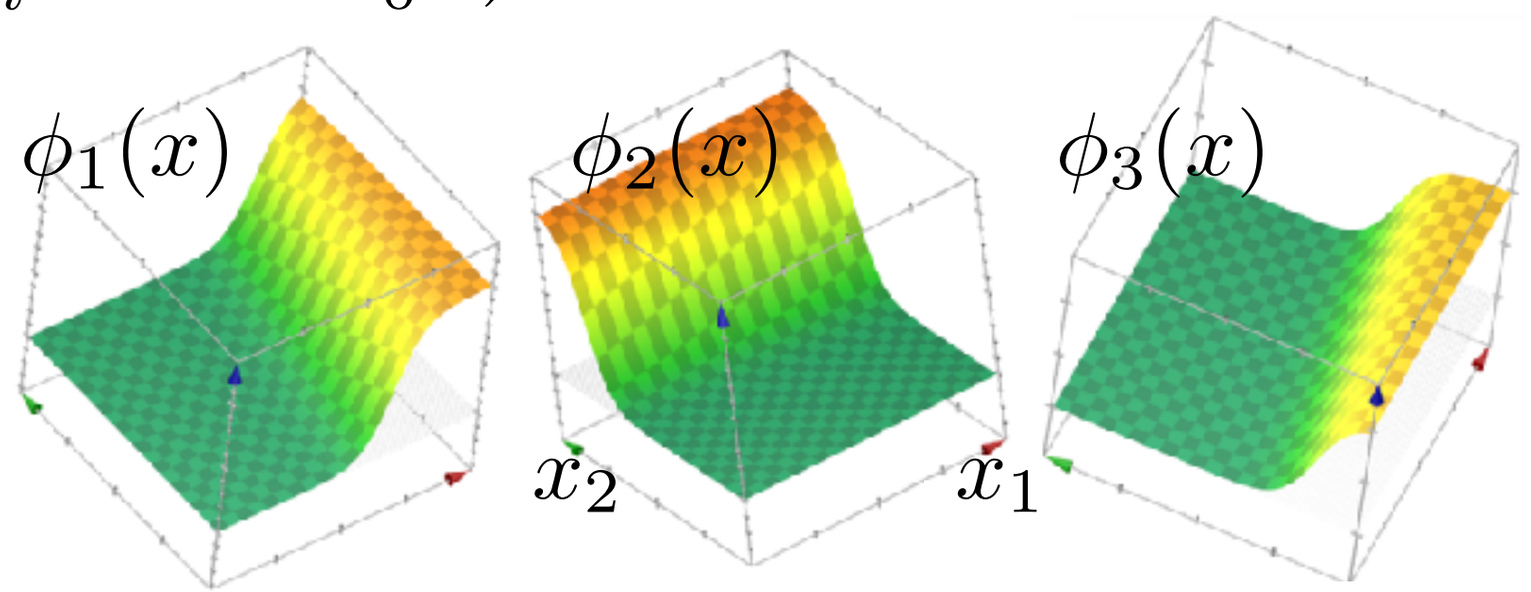


- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = z$
  - Choose $f^{(2)}(z) = \sigma(z)$

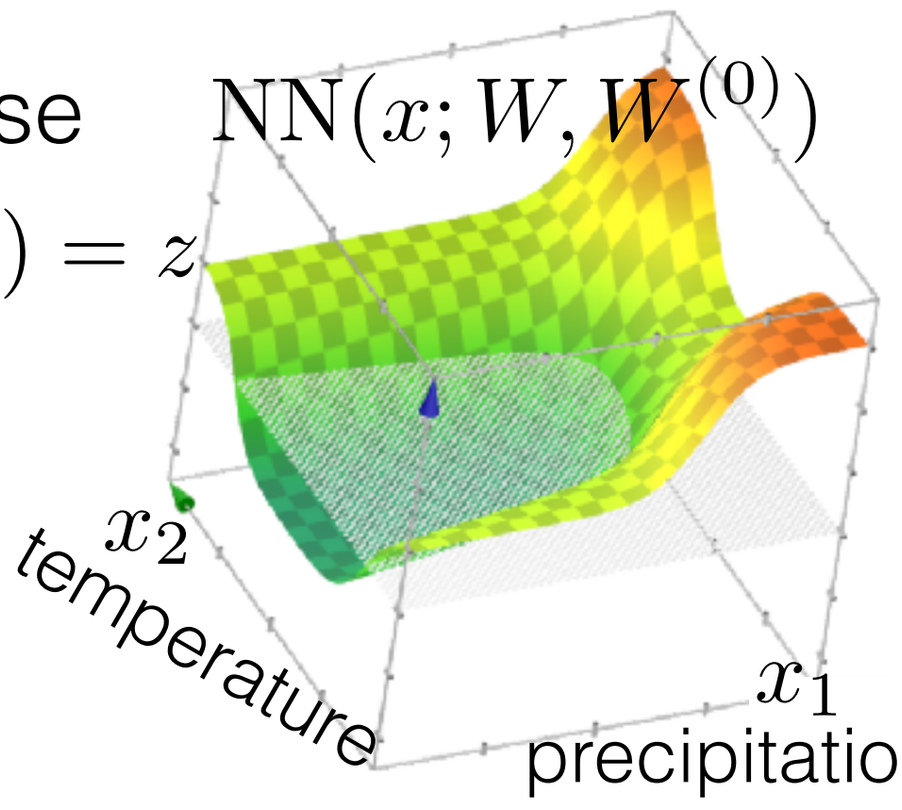


Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$

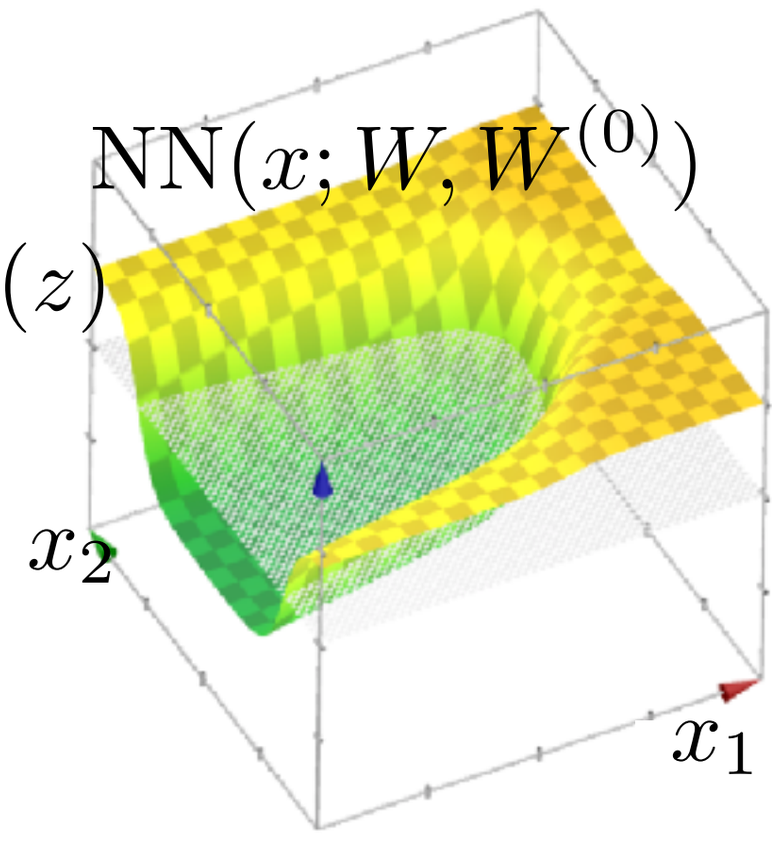


- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = z$



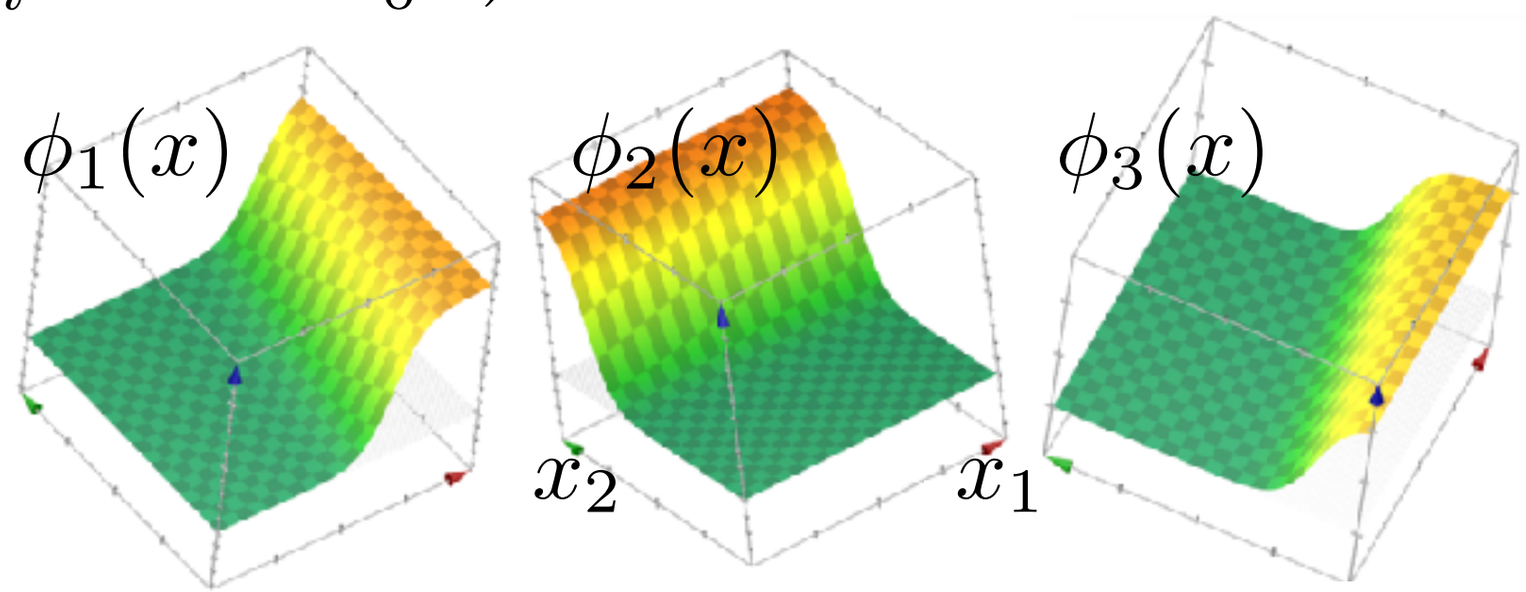
How much will I dislike running today?

- Choose $f^{(2)}(z) = \sigma(z)$



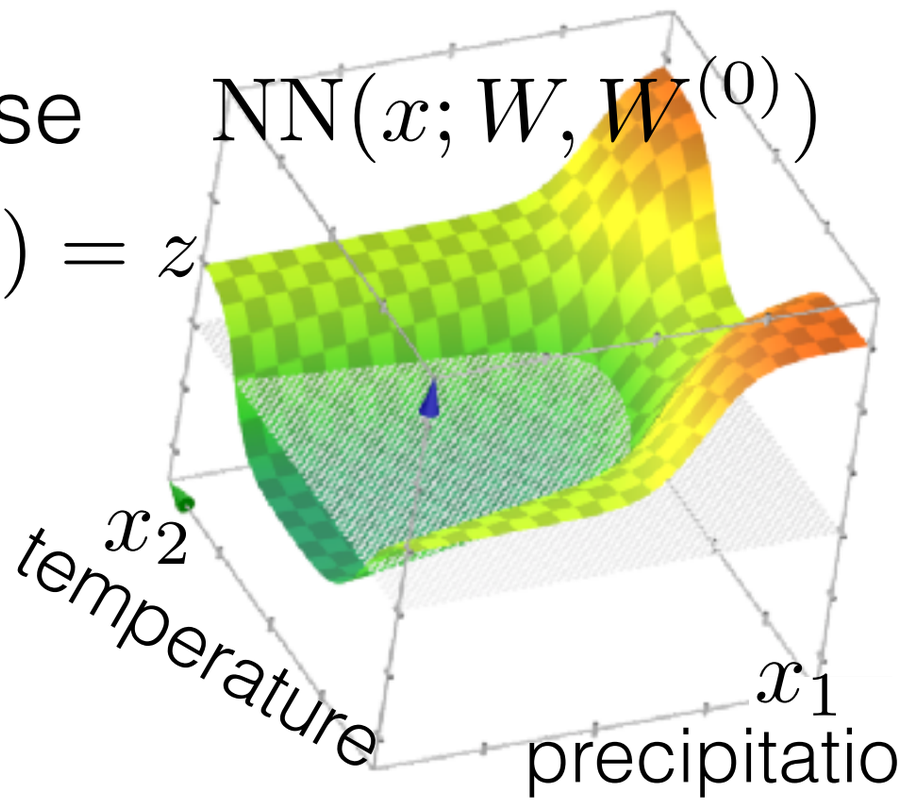
Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$

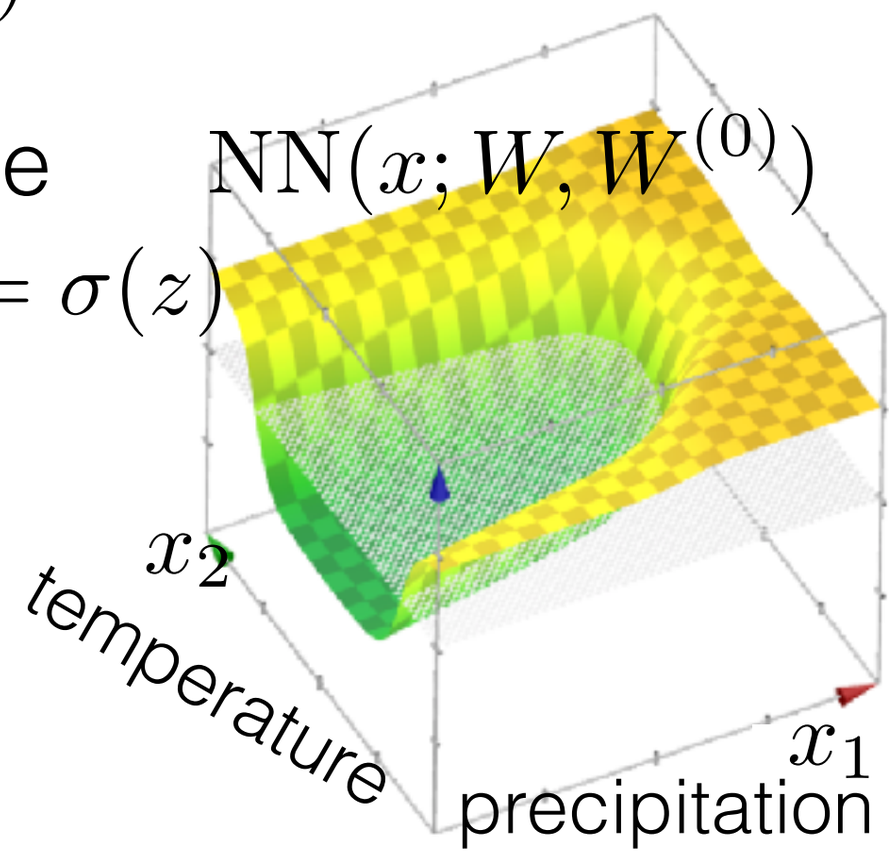


- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = z$

How much will I dislike running today?

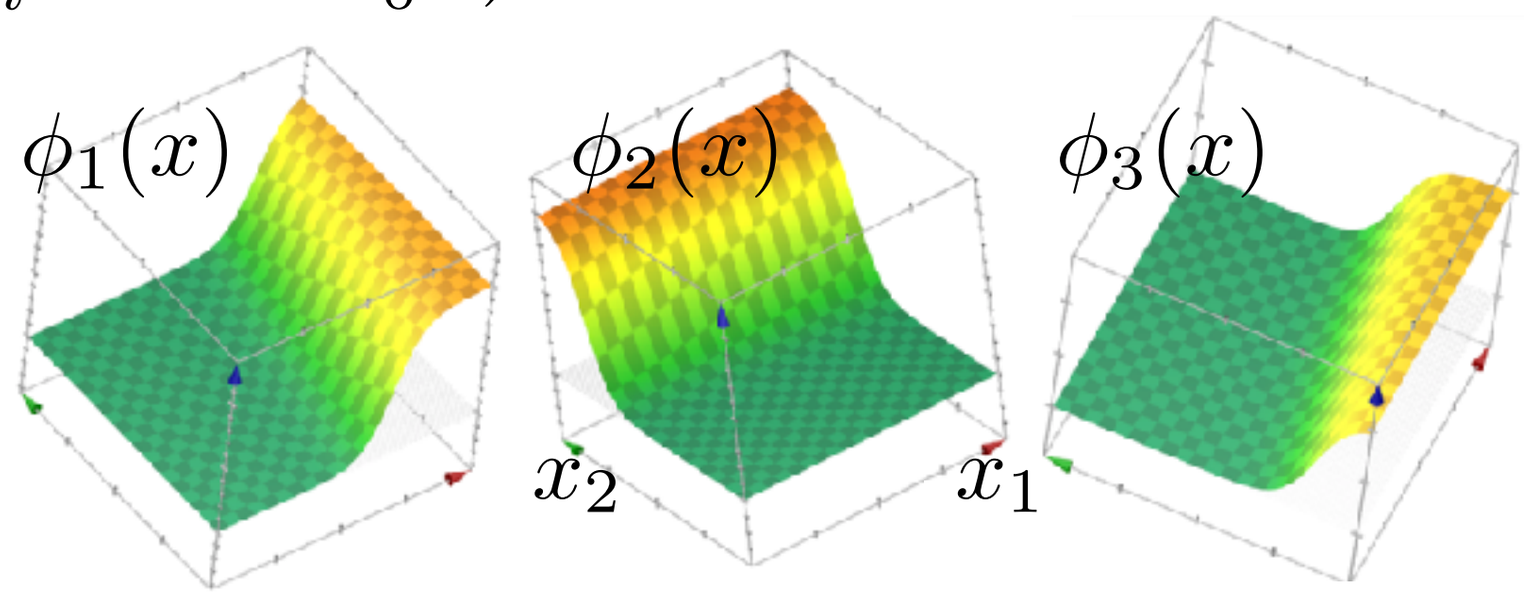


- Choose $f^{(2)}(z) = \sigma(z)$



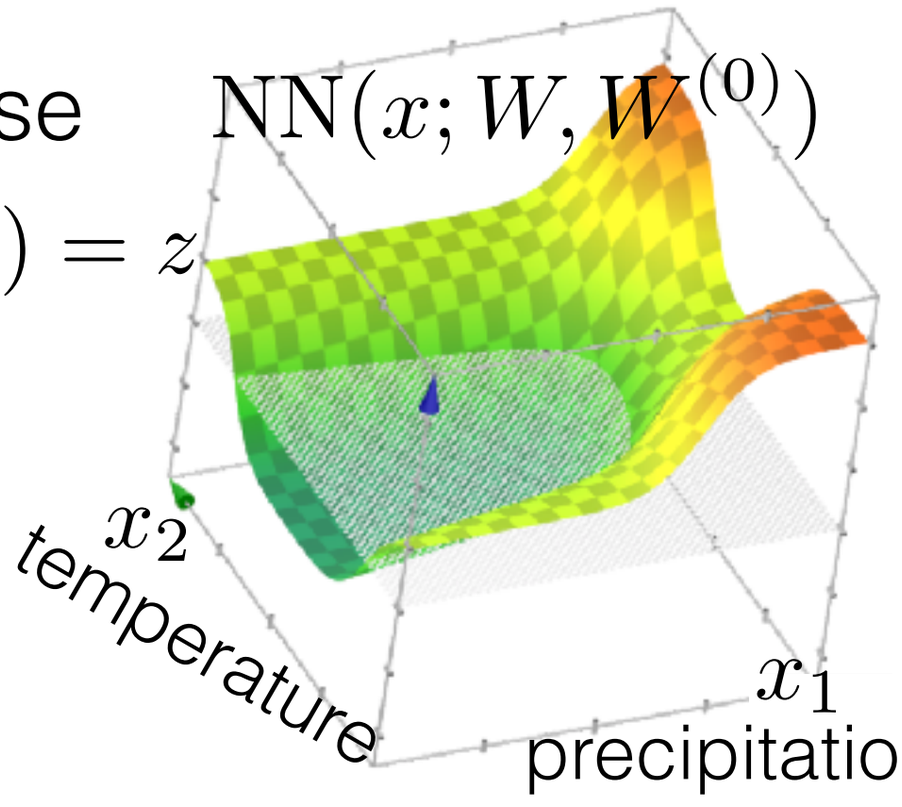
Choices of activation function

- 1st layer: $A_i^{(1)} = f^{(1)}(w_i^{(1)\top} x + w_0^{(1)})$
  - Choose $f^{(1)}(z) = \sigma(z)$



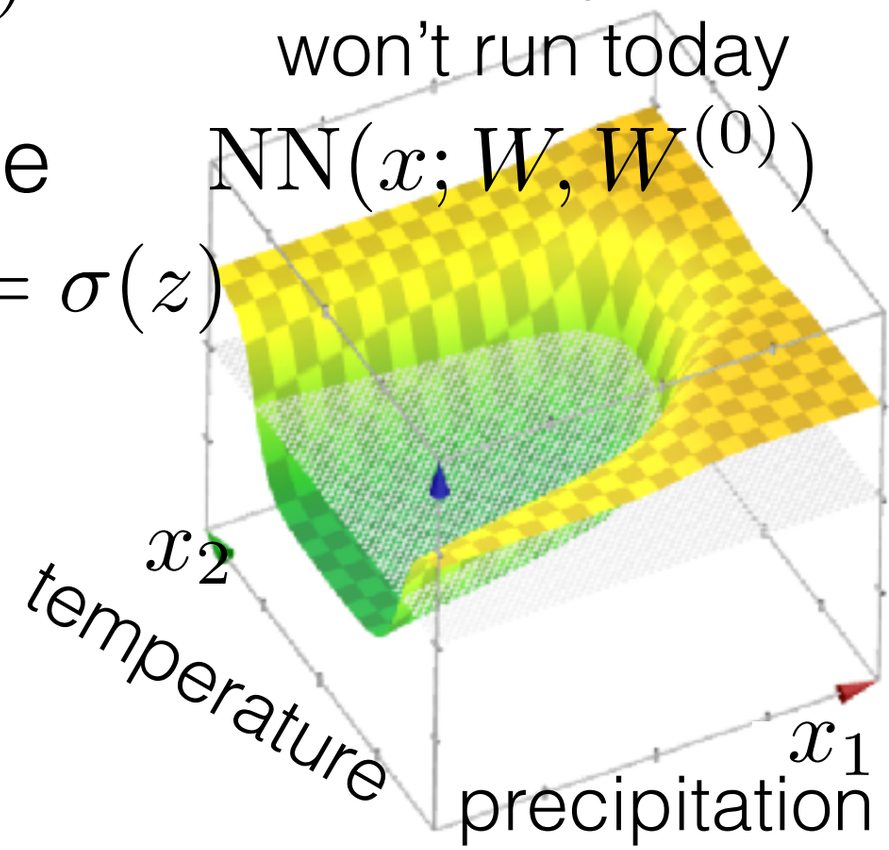
- 2nd layer: $A_i^{(2)} = f^{(2)}(w_i^{(2)\top} A^{(1)} + w_0^{(2)})$
  - Choose $f^{(2)}(z) = z$

How much will I dislike running today?



- Choose $f^{(2)}(z) = \sigma(z)$

Probability that I won't run today



Learning the parameters

Learning the parameters

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})$

Learning the parameters

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$

Learning the parameters

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss$_i$ w.r.t. parameters

Learning the parameters

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss$_i$ w.r.t. parameters

w.r.t. = with respect to

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss$_i$ w.r.t. parameters

w.r.t. = with respect to

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1 $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
    - Example: $\partial \text{loss}_i / \partial W^{(1)}_{1,1}$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u

  - Example: $\partial \underbrace{\text{loss}_i}_{\text{1x1}} / \partial W^{(1)}_{1,1}$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1 $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\partial \text{loss}_i / \partial W^{(1)}_{1,1}$ (1x1 / 1x1)

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1 $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\partial \text{loss}_i / \partial W^{(1)}_{1,1}$ (1x1 / 1x1; 1x1)

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1
  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
    - Example: $\partial \text{loss}_i / \partial W^{(1)}_{1,1}$ (1x1 / 1x1 / 1x1)
- Convention: for v of size 1×1 and u of size $b \times 1$
  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\partial \text{loss}_i / \partial W^{(1)}_{1,1}$ (1x1 / 1x1 / 1x1)
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\partial \text{loss}_i / \partial W^{(1)}_0$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\partial \text{loss}_i / \partial W^{(1)}_{1,1}$ (1x1 / 1x1 / 1x1)
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\partial \text{loss}_i / \partial W^{(1)}_0 = \nabla_{W^{(1)}_0} \text{loss}_i$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u

  - Example: $\partial \text{loss}_i / \partial W_{1,1}^{(1)}$ (1x1 / 1x1, 1x1)
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$

  - Example: $\partial \text{loss}_i / \partial W_0^{(1)} = \nabla_{W_0^{(1)}} \text{loss}_i$ (1x1)

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0} = \nabla_{W^{(1)}_0} \text{loss}_i$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\partial A^{(1)} / \partial Z^{(1)}$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\underset{n^1\text{x}1}{\partial A^{(1)}} / \partial Z^{(1)}$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\underset{n^1\text{x}1}{\partial A^{(1)}} / \underset{n^1\text{x}1}{\partial Z^{(1)}}$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\underset{n^1\text{x}1}{\partial A^{(1)}} / \underset{n^1\text{x}1}{\partial Z^{(1)}}$ n^1xn^1

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\underset{n^1\text{x}1}{\partial A^{(1)}} / \underset{n^1\text{x}1}{\partial Z^{(1)}}$ n^1xn^1
- Convention: for v of size 1×1 and u of size $b \times c$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v / \partial u_{j,k}$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\underset{n^1\text{x}1}{\partial A^{(1)}} / \underset{n^1\text{x}1}{\partial Z^{(1)}}$ n^1xn^1
- Convention: for v of size 1×1 and u of size $b \times c$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v / \partial u_{j,k}$
  - Example: $\partial \text{loss}_i / \partial W^{(1)}$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\underset{n^1\text{x}1}{\partial A^{(1)}} / \underset{n^1\text{x}1}{\partial Z^{(1)}}$ n^1xn^1
- Convention: for v of size 1×1 and u of size $b \times c$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v / \partial u_{j,k}$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \partial W^{(1)}$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\underset{n^1\text{x}1}{\partial A^{(1)}} / \underset{n^1\text{x}1}{\partial Z^{(1)}}$ n^1xn^1
- Convention: for v of size 1×1 and u of size $b \times c$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v / \partial u_{j,k}$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{m^1\text{x}n^1}{\partial W^{(1)}}$

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n_{\text{data}}} \sum_{i=1}^{n_{\text{data}}} \overbrace{L(\text{NN}(x^{(i)}; W, W_0), y^{(i)})}^{\text{loss}_i}$
- For SGD, we'll want derivatives of loss_i w.r.t. parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{1\text{x}1}{\partial W^{(1)}_{1,1}}$ 1x1
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{n^1\text{x}1}{\partial W^{(1)}_0}$ n^1x1 $= \nabla_{W^{(1)}_0} \text{loss}_i$
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\underset{n^1\text{x}1}{\partial A^{(1)}} / \underset{n^1\text{x}1}{\partial Z^{(1)}}$ n^1xn^1
- Convention: for v of size 1×1 and u of size $b \times c$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v / \partial u_{j,k}$
  - Example: $\underset{1\text{x}1}{\partial \text{loss}_i} / \underset{m^1\text{x}n^1}{\partial W^{(1)}}$ m^1xn^1

Derivatives: Notation

- Objective: $J(W, W_0) = \frac{1}{n}$ loss_i
- For SGD, we'll want deri[illegible] parameters
- Convention: for v of size 1×1 and u of size 1×1

  $\partial v / \partial u$ is the (scalar) partial derivative of v w.r.t. u
  - Example: $\partial \text{loss}_i / \partial W^{(1)}_{1,1}$ (1x1 / 1x1; 1x1)
- Convention: for v of size 1×1 and u of size $b \times 1$

  $\partial v / \partial u$ is a vector of size $b \times 1$ with jth entry $\partial v / \partial u_j$
  - Example: $\partial \text{loss}_i / \partial W^{(1)}_0 = \nabla_{W^{(1)}_0} \text{loss}_i$ (1x1 / n^1x1; n^1x1)
- Convention: for v of size $c \times 1$ and u of size $b \times 1$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v_k / \partial u_j$
  - Example: $\partial A^{(1)} / \partial Z^{(1)}$ (n^1x1 / n^1x1; n^1xn^1)
- Convention: for v of size 1×1 and u of size $b \times c$

  $\partial v / \partial u$ is a matrix of size $b \times c$ with (j, k) entry $\partial v / \partial u_{j,k}$
  - Example: $\partial \text{loss}_i / \partial W^{(1)}$ (1x1 / m^1xn^1; m^1xn^1)

Exercise: check that when these conventions overlap, they agree

Taking derivatives for SGD

Taking derivatives for SGD

- Example 1-layer NN

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = W^{(1)\top} x + W_0^{(1)}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = W^{(1)\top} x + W_0^{(1)}$$

4x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = W^{(1)\top} x + W_0^{(1)}$$

3x1 4x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{4\text{x}1}$$

3x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

$$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step t

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

notice: we're dropping *i*

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

notice: we're dropping *i*

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1

notice: we're dropping *i*

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1 1x1

notice: we're dropping *i*

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1 1x1 1x1

notice: we're dropping *i*

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1 1x1 1x1

notice: we're dropping *i*

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x3 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1 1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1 1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x3 4x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x3 4x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} =$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} =$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} =$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} =$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x3 4x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x3 4x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$\underbrace{W_{0,j}^{(1)}}_{1\text{x}1} \leftarrow \underbrace{W_{0,j}^{(1)}}_{1\text{x}1} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x3 4x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}$$

1x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x3 4x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1 1x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

1x1 1x1 1x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

1x1 1x1 1x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

1x1 1x1 1x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

1x1 1x1 1x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

1x1 1x1 1x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x3 4x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$\underset{1\text{x}1}{W_{0,j}^{(1)}} \leftarrow \underset{1\text{x}1}{W_{0,j}^{(1)}} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \frac{\partial \text{loss}}{\partial A^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$\underset{1\text{x}1}{W_{0,j}^{(1)}} \leftarrow \underset{1\text{x}1}{W_{0,j}^{(1)}} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$\underset{1\text{x}1}{W_{0,j}^{(1)}} \leftarrow \underset{1\text{x}1}{W_{0,j}^{(1)}} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$\underset{1\text{x}1}{W_{0,j}^{(1)}} \leftarrow \underset{1\text{x}1}{W_{0,j}^{(1)}} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

Notice: order of the terms

$$\underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$\underset{1\text{x}1}{W_{0,j}^{(1)}} \leftarrow \underset{1\text{x}1}{W_{0,j}^{(1)}} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

Notice: order of the terms

$$\underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}}$$

Exercise: check that you can't just switch the order and get the same result

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}}$$

$$\frac{\partial \text{loss}}{\partial W_0^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}}$$

$$\underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial W_0^{(1)}}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

1x1 1x1 1x1

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A^{(1)}} \qquad \frac{\partial \text{loss}}{\partial W_0^{(1)}}$$

1x1 1x3 3x1 3x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}} \qquad \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial W_0^{(1)}}} = \frac{\partial A^{(1)}}{\partial W_0^{(1)}} \frac{\partial \text{loss}}{\partial A^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}} \qquad \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial W_0^{(1)}}} = \underset{3\text{x}3}{\frac{\partial A^{(1)}}{\partial W_0^{(1)}}} \frac{\partial \text{loss}}{\partial A^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + \underbrace{W_0^{(1)}}_{3\text{x}1}$$

3x1 4x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial A_k^{(1)}}} \underset{1\text{x}1}{\frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

$$\underset{1\text{x}1}{\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}} = \underset{1\text{x}3}{\frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}} \qquad \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial W_0^{(1)}}} = \underset{3\text{x}3}{\frac{\partial A^{(1)}}{\partial W_0^{(1)}}} \underset{3\text{x}1}{\frac{\partial \text{loss}}{\partial A^{(1)}}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

1x1 1x1 1x1

Exercise: compute this matrix

$$\frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}} = \frac{\partial A^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A^{(1)}} \qquad \frac{\partial \text{loss}}{\partial W_0^{(1)}} = \frac{\partial A^{(1)}}{\partial W_0^{(1)}} \frac{\partial \text{loss}}{\partial A^{(1)}}$$

1x1 1x3 3x1 3x1 3x3 3x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Part of SGD step *t*: randomly choose (*x*,*y*) and update:

value at end of step *t*

$$W_{0,j}^{(1)} \leftarrow W_{0,j}^{(1)} - \eta(t) \frac{\partial \text{loss}}{\partial W_{0,j}^{(1)}}$$

1x1 1x1

notice: we're dropping *i*

all values on this side are from step *t*-1

- Use the scalar chain rule from calculus:

$$\frac{\partial \text{loss}}{\partial W^{(1)}} = \sum_{k=1}^{3} \frac{\partial \text{loss}}{\partial A_k^{(1)}} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} = \sum_{k=1}^{3} \frac{\partial A_k^{(1)}}{\partial W_{0,j}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

Exercise: compute this matrix

Notice: We can't just substitute the weight **matrix** in the final formula. (Why not?)

$$\frac{\partial \text{loss}}{\partial W_0^{(1)}} = \frac{\partial A^{(1)}}{\partial W_0^{(1)}} \frac{\partial \text{loss}}{\partial A^{(1)}}$$

3x1 3x3 3x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x3 4x1 3x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$

$$x \left(\frac{\partial \text{loss}}{\partial A^{(1)}} \right)^\top$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)^\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$

$$x \left(\frac{\partial \text{loss}}{\partial A^{(1)}} \right)^\top$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x3 4x1 3x1

1x1 $$\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$
- By our convention, $\partial \text{loss} / \partial W^{(1)}$ is a matrix with (j,k) entry $\partial \text{loss} / \partial W_{j,k}^{(1)}$

$$x \left(\frac{\partial \text{loss}}{\partial A^{(1)}} \right)^\top$$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x3 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$
- By our convention, $\partial \text{loss} / \partial W^{(1)}$ is a matrix with (j,k) entry $\partial \text{loss} / \partial W_{j,k}^{(1)}$
- So: $\frac{\partial \text{loss}}{\partial W^{(1)}} = x \left(\frac{\partial \text{loss}}{\partial A^{(1)}} \right)^\top$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x3 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$
- By our convention, $\partial \text{loss} / \partial W^{(1)}$ is a matrix with (j,k) entry $\partial \text{loss} / \partial W_{j,k}^{(1)}$
  - So: $\frac{\partial \text{loss}}{\partial W^{(1)}} = x \left(\frac{\partial \text{loss}}{\partial A^{(1)}} \right)^\top$

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$
- By our convention, $\partial \text{loss} / \partial W^{(1)}$ is a matrix with (j,k) entry $\partial \text{loss} / \partial W_{j,k}^{(1)}$
- So: $\frac{\partial \text{loss}}{\partial W^{(1)}} = x \left(\frac{\partial \text{loss}}{\partial A^{(1)}} \right)^\top$

4x3

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x3 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$
- By our convention, $\partial \text{loss} / \partial W^{(1)}$ is a matrix with (j,k) entry $\partial \text{loss} / \partial W_{j,k}^{(1)}$
- So: $\frac{\partial \text{loss}}{\partial W^{(1)}} = x \left(\frac{\partial \text{loss}}{\partial A^{(1)}} \right)^\top$

4x3 4x1

Taking derivatives for SGD

- Example 1-layer NN
  - data dimension $m^{(1)} = 4$; # outputs $n^{(1)} = 3$

$$A^{(1)} = \underbrace{W^{(1)\top}}_{4\text{x}3} x + W_0^{(1)}$$

3x1 4x3 4x1 3x1

1x1 $\text{loss} = L(A^{(1)}, y) = \sum_{k=1}^{3} (A_k^{(1)} - y_k)^2$

- Our last derivation doesn't work for a full weight matrix. (Exercise: why not?) So what about the full matrix case?

1x1 $$\frac{\partial \text{loss}}{\partial W_{j,k}^{(1)}} = \sum_{p=1}^{3} \frac{\partial A_p^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_p^{(1)}} = \frac{\partial A_k^{(1)}}{\partial W_{j,k}^{(1)}} \frac{\partial \text{loss}}{\partial A_k^{(1)}} = x_j \frac{\partial \text{loss}}{\partial A_k^{(1)}}$$

- Observe: if u is a $c \times 1$ vector and v is a $b \times 1$ vector, $vu^\top$ is a matrix of size $b \times c$ with (j,k) entry $v_j u_k$
- By our convention, $\partial \text{loss} / \partial W^{(1)}$ is a matrix with (j,k) entry $\partial \text{loss} / \partial W_{j,k}^{(1)}$
- So: $$\underset{4\text{x}3}{\frac{\partial \text{loss}}{\partial W^{(1)}}} = \underset{4\text{x}1}{x} \underbrace{\left(\frac{\partial \text{loss}}{\partial A^{(1)}} \right)^\top}_{3\text{x}1}$$

Taking derivatives for SGD

Taking derivatives for SGD

- General case: NN with *L* layers

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)}$$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss} / \partial W^{(\ell)}$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss} / \partial W^{(\ell)}$, break off the “final” weight derivative

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss} / \partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} =$$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss} / \partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}}$$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss} / \partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}}$$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss} / \partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}}$$

1x1 1x1 1x1 1x1 1x1

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss} / \partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}$$

1x1 1x1 1x1 1x1 1x1

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss} / \partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss} / \partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$ $1 \times n^{\ell}$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$ $1 \times n^{\ell}$

- It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}}\right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 m^{ℓ}xn^{ℓ} m^{ℓ}x1 1xn^{ℓ}

  - It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} =$$

n^{ℓ}x1

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}}\right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 m^{ℓ}xn^{ℓ} m^{ℓ}x1 1xn^{ℓ}

  - It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \qquad \frac{\partial \text{loss}}{\partial A^{(L)}}$$

n^{ℓ}x1 n^{L}x1

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}}\right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 m^{ℓ}xn^{ℓ} m^{ℓ}x1 1xn^{ℓ}

  - It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \cdot \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

n^{ℓ}x1 n^{L}xn^{L} n^{L}x1

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}}\right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 m^{ℓ}xn^{ℓ} m^{ℓ}x1 1xn^{ℓ}

  - It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

n^{ℓ}x1 n^{L-1}xn^{L} n^{L}xn^{L} n^{L}x1

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}}\right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 m^{ℓ}xn^{ℓ} m^{ℓ}x1 1xn^{ℓ}

  - It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial A^{(L-1)}}{\partial Z^{(L-1)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

n^{ℓ}x1 n^{L-1}xn^{L-1} n^{L-1}xn^{L} n^{L}xn^{L} n^{L}x1

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$ $1 \times n^{\ell}$

- It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial A^{(\ell+1)}}{\partial Z^{(\ell+1)}} \cdots \frac{\partial A^{(L-1)}}{\partial Z^{(L-1)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

$n^{\ell} \times 1$ $n^{\ell+1} \times n^{\ell+1}$ $n^{L-1} \times n^{L-1}$ $n^{L-1} \times n^{L}$ $n^{L} \times n^{L}$ $n^{L} \times 1$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$ $1 \times n^{\ell}$

- It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial Z^{(\ell+1)}}{\partial A^{(\ell)}} \frac{\partial A^{(\ell+1)}}{\partial Z^{(\ell+1)}} \cdots \frac{\partial A^{(L-1)}}{\partial Z^{(L-1)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

$n^{\ell} \times 1$ $n^{\ell} \times n^{\ell+1}$ $n^{\ell+1} \times n^{\ell+1}$ $n^{L-1} \times n^{L-1}$ $n^{L-1} \times n^{L}$ $n^{L} \times n^{L}$ $n^{L} \times 1$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$ $1 \times n^{\ell}$

- It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial A^{(\ell)}}{\partial Z^{(\ell)}} \frac{\partial Z^{(\ell+1)}}{\partial A^{(\ell)}} \frac{\partial A^{(\ell+1)}}{\partial Z^{(\ell+1)}} \cdots \frac{\partial A^{(L-1)}}{\partial Z^{(L-1)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

$n^{\ell} \times 1$ $n^{\ell} \times n^{\ell}$ $n^{\ell} \times n^{\ell+1}$ $n^{\ell+1} \times n^{\ell+1}$ $n^{L-1} \times n^{L-1}$ $n^{L-1} \times n^{L}$ $n^{L} \times n^{L}$ $n^{L} \times 1$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}}\right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$ $1 \times n^{\ell}$

  - It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial A^{(\ell)}}{\partial Z^{(\ell)}} \frac{\partial Z^{(\ell+1)}}{\partial A^{(\ell)}} \frac{\partial A^{(\ell+1)}}{\partial Z^{(\ell+1)}} \cdots \frac{\partial A^{(L-1)}}{\partial Z^{(L-1)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

$n^{\ell} \times 1$ $n^{\ell} \times n^{\ell}$ $n^{\ell} \times n^{\ell+1}$ $n^{\ell+1} \times n^{\ell+1}$ $n^{L-1} \times n^{L-1}$ $n^{L-1} \times n^{L}$ $n^{L} \times n^{L}$ $n^{L} \times 1$

    - Lots of this computation will be shared across layers

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$ $1 \times n^{\ell}$

- It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial A^{(\ell)}}{\partial Z^{(\ell)}} \frac{\partial Z^{(\ell+1)}}{\partial A^{(\ell)}} \frac{\partial A^{(\ell+1)}}{\partial Z^{(\ell+1)}} \cdots \frac{\partial A^{(L-1)}}{\partial Z^{(L-1)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

$n^{\ell} \times 1$ $n^{\ell} \times n^{\ell}$ $n^{\ell} \times n^{\ell+1}$ $n^{\ell+1} \times n^{\ell+1}$ $n^{L-1} \times n^{L-1}$ $n^{L-1} \times n^{L}$ $n^{L} \times n^{L}$ $n^{L} \times 1$

  - Lots of this computation will be shared across layers
  - More efficient to compute the shared parts only once

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)} = m^{(l+1)}$ outputs

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$ $1 \times n^{\ell}$

  - It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial A^{(\ell)}}{\partial Z^{(\ell)}} \frac{\partial Z^{(\ell+1)}}{\partial A^{(\ell)}} \frac{\partial A^{(\ell+1)}}{\partial Z^{(\ell+1)}} \cdots \frac{\partial A^{(L-1)}}{\partial Z^{(L-1)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

$n^{\ell} \times 1$ $n^{\ell} \times n^{\ell}$ $n^{\ell} \times n^{\ell+1}$ $n^{\ell+1} \times n^{\ell+1}$ $n^{L-1} \times n^{L-1}$ $n^{L-1} \times n^{L}$ $n^{L} \times n^{L}$ $n^{L} \times 1$

    - Lots of this computation will be shared across layers
    - More efficient to compute the shared parts only once

- Can similarly show: $\partial \text{loss}/\partial W_0^{(\ell)} = \partial \text{loss}/\partial Z^{(\ell)}$

Taking derivatives for SGD

- General case: NN with *L* layers
  - Layer ℓ has $m^{(l)}$ inputs and $n^{(l)}$ outputs

Exercise: Check that you agree with these formulas!

$$A^{(\ell)} = f^{\ell}(Z^{(\ell)}), Z^{(\ell)} = W^{(\ell)\top} A^{(\ell-1)} + W_0^{(\ell)} \quad \text{with} \quad A^{(0)} = x$$

- To find $\partial \text{loss}/\partial W^{(\ell)}$, break off the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \Rightarrow \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}}\right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 m^{ℓ}xn^{ℓ} m^{ℓ}x1 1xn^{ℓ}

  - It remains to find $\partial \text{loss}/\partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial A^{(\ell)}}{\partial Z^{(\ell)}} \frac{\partial Z^{(\ell+1)}}{\partial A^{(\ell)}} \frac{\partial A^{(\ell+1)}}{\partial Z^{(\ell+1)}} \cdots \frac{\partial A^{(L-1)}}{\partial Z^{(L-1)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

n^{ℓ}x1 n^{ℓ}xn^{ℓ} n^{ℓ}x$n^{\ell+1}$ $n^{\ell+1}$x$n^{\ell+1}$ n^{L-1}xn^{L-1} n^{L-1}xn^{L} n^{L}xn^{L} n^{L}x1

    - Lots of this computation will be shared across layers
    - More efficient to compute the shared parts only once

- Can similarly show: $\partial \text{loss}/\partial W_0^{(\ell)} = \partial \text{loss}/\partial Z^{(\ell)}$

Taking derivatives for SGD

Exercise: use these formulas to show that the final derivatives w.r.t. W^{ℓ} are 0 if any layer above ℓ has step function activations

Exercise: Check that you agree with these formulas!

- ers
- nd
- $A^{(\ell-1)} + W_0^{(\ell)}$ with $A^{(0)} = x$
- ff the "final" weight derivative

$$\frac{\partial \text{loss}}{\partial W_{j,k}^{(\ell)}} = \frac{\partial Z_k^{(\ell)}}{\partial W_{j,k}^{(\ell)}} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} = A_j^{(\ell-1)} \frac{\partial \text{loss}}{\partial Z_k^{(\ell)}} \quad \Rightarrow \quad \frac{\partial \text{loss}}{\partial W^{(\ell)}} = A^{(\ell-1)} \underbrace{\left(\frac{\partial \text{loss}}{\partial Z^{(\ell)}} \right)^{\top}}$$

1x1 1x1 1x1 1x1 1x1 $m^{\ell} \times n^{\ell}$ $m^{\ell} \times 1$ $1 \times n^{\ell}$

- It remains to find $\partial \text{loss} / \partial Z^{(\ell)}$

$$\frac{\partial \text{loss}}{\partial Z^{(\ell)}} = \frac{\partial A^{(\ell)}}{\partial Z^{(\ell)}} \frac{\partial Z^{(\ell+1)}}{\partial A^{(\ell)}} \frac{\partial A^{(\ell+1)}}{\partial Z^{(\ell+1)}} \cdots \frac{\partial A^{(L-1)}}{\partial Z^{(L-1)}} \frac{\partial Z^{(L)}}{\partial A^{(L-1)}} \frac{\partial A^{(L)}}{\partial Z^{(L)}} \frac{\partial \text{loss}}{\partial A^{(L)}}$$

$n^{\ell} \times 1$ $n^{\ell} \times n^{\ell}$ $n^{\ell} \times n^{\ell+1}$ $n^{\ell+1} \times n^{\ell+1}$ $n^{L-1} \times n^{L-1}$ $n^{L-1} \times n^{L}$ $n^{L} \times n^{L}$ $n^{L} \times 1$

  - Lots of this computation will be shared across layers
  - More efficient to compute the shared parts only once

- Can similarly show: $\partial \text{loss} / \partial W_0^{(\ell)} = \partial \text{loss} / \partial Z^{(\ell)}$