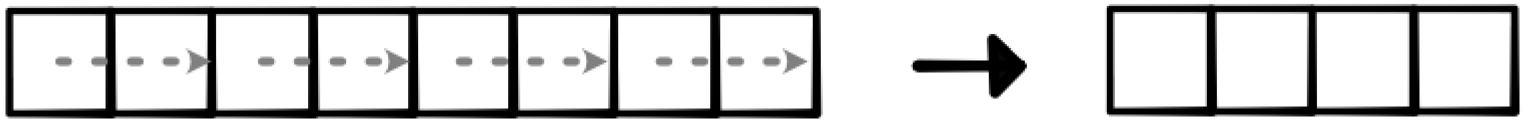
Processing math: 7%

Module 4.3 - Advanced NNs

Processing math: 7%

"Pooling"

Reduction applied to each region:

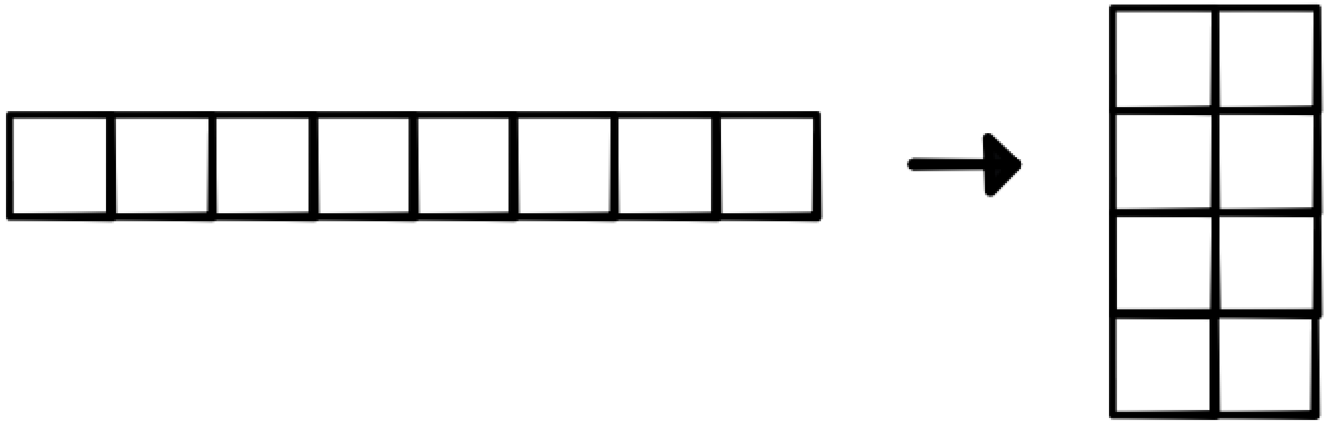


Processing math: 7%

Processing math: 7%

Simple Implementation

- Ensure that it is contiguous
- Use View to "fold" the tensor

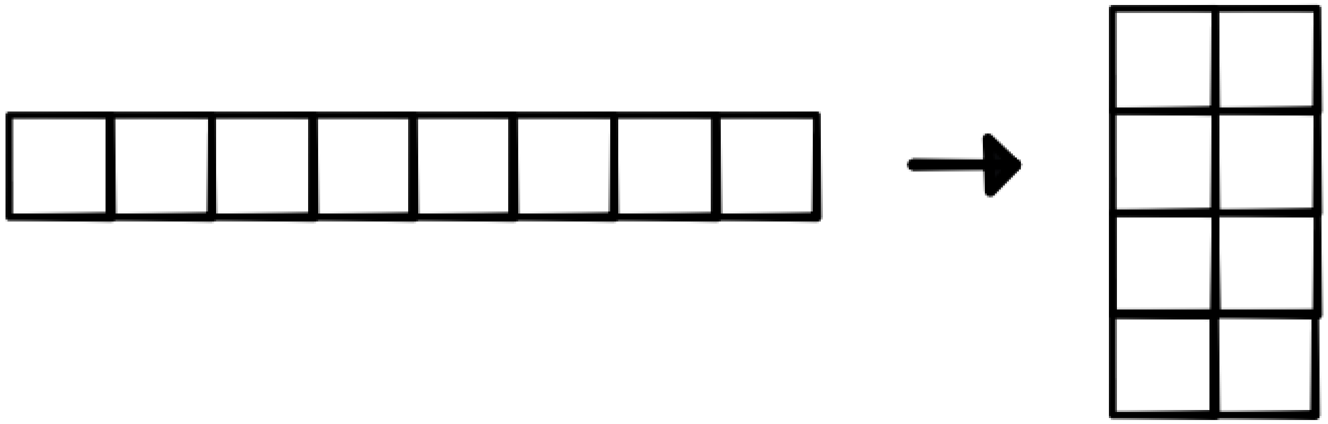


Processing math: 7%

Processing math: 7%

Why does folding work?

- View requires "contiguous" tensor
- View(4, 2) makes strides (2, 1)

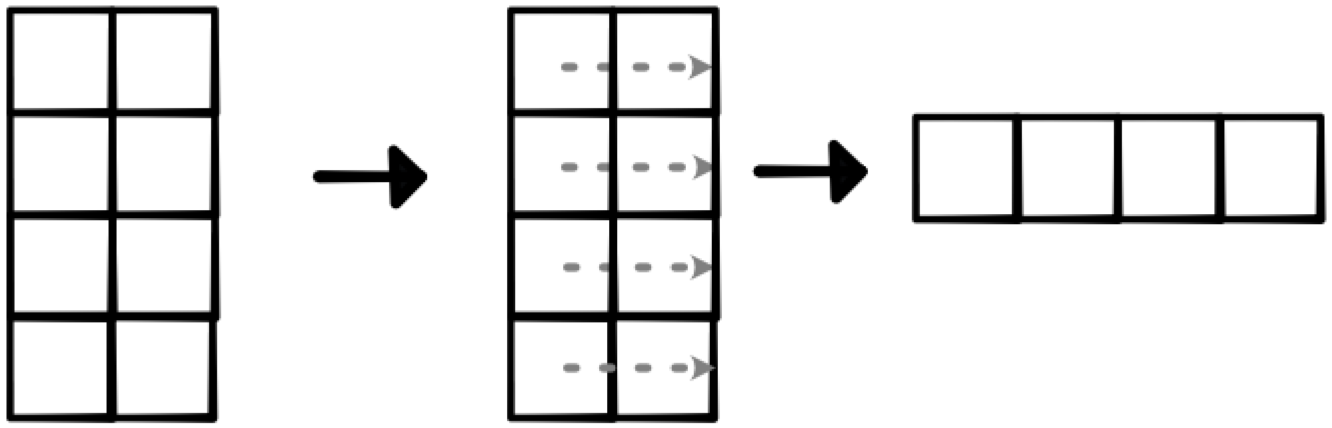


Processing math: 7%

Processing math: 7%

Simple Implementation

- Reduce along created fold



Processing math: 7%

Processing math: 7%

Quiz

Processing math: 7%

Processing math: 7%

Gradient Flow

- Layers that are used get more updates
- Gradient signals which aspect was important
- Can have extra layers

Processing math: 7%

More Reductions

- Heading for a `max` reduction
- Heading for a `softmax` output
- Quick detour

Processing math: 7%

Processing math: 7%

ReLU, Step, Sigmoid

Processing math: 7%

Processing math: 7%

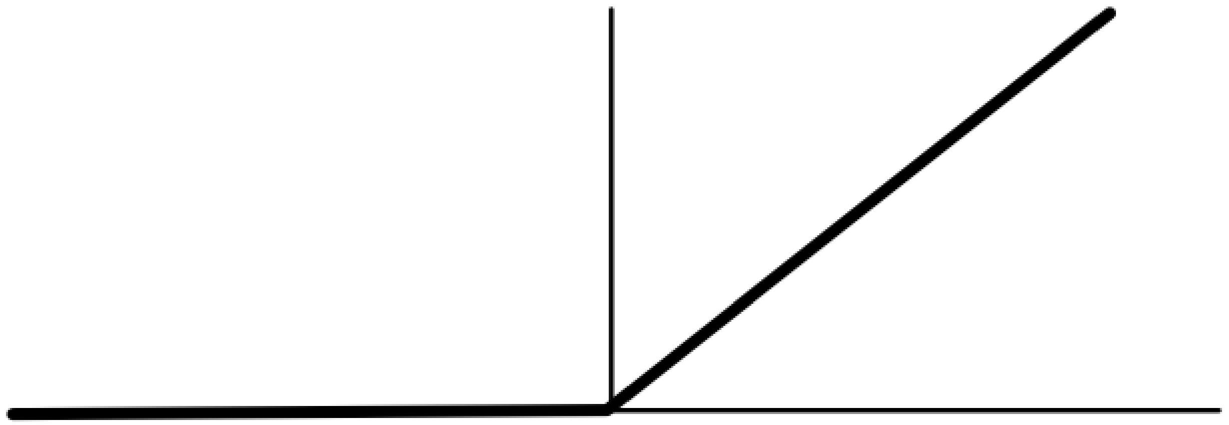
Basic Operations

- Introduced in Module-0
- Widely used in ML
- What is it?

Processing math: 7%

Simple Function: ReLU

Main "activation" function



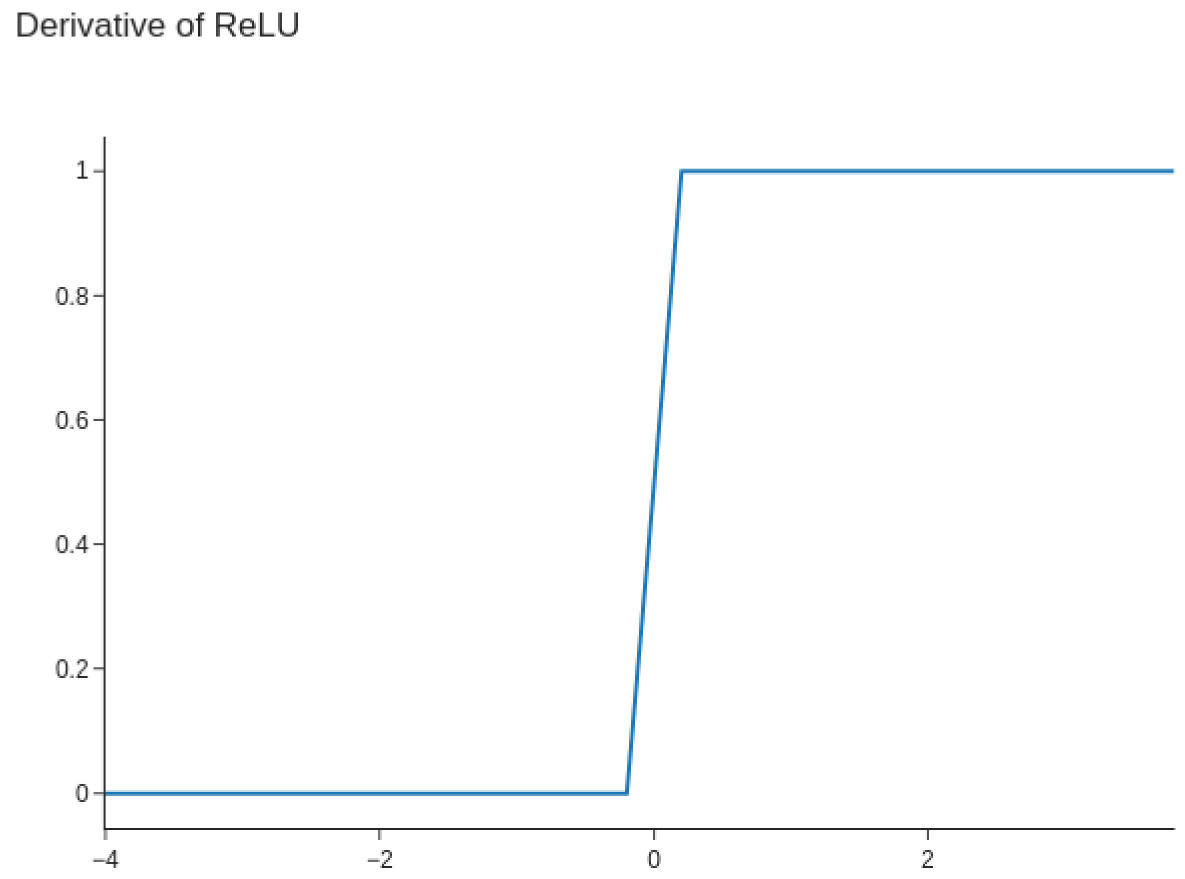
Primarily used to split the data.

Processing math: 7%

Processing math: 7%

Simple Function: Step

Step function $f(x) = x > 0$ determines correct answer



Processing math: 7%

Processing math: 7%

ReLU

Mathematically,

$\mathrm{ReLU}(x) = \max$

Simplest `max` function.

Processing math: 7%

Processing math: 7%

Step

Mathematically,

$$\text{step}(x) = x > 0 = \arg\max\{0, x\}$$

Simplest `argmax` function.

Processing math: 7%

Relationship

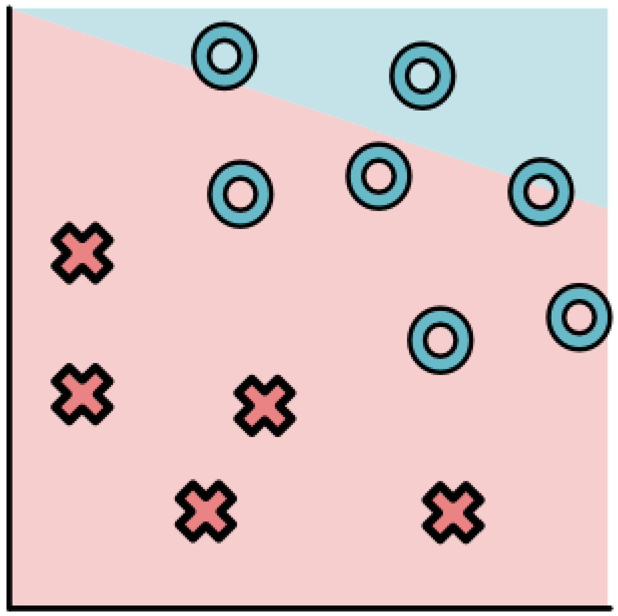
Step is derivative of ReLU

$$\mathrm{ReLU}'(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{ow} \end{cases}$$

$$\mathrm{step}(x) = \mathrm{ReLU}'(x)$$

Processing math: 7%

Processing math: 7%



Loss of step tells us how many points are wrong.

Processing math: 7%

Processing math: 7%

Derivative of Step?

Mathematically,

\text{step}'(x) =

$$\begin{cases} 0 & \text{if } x \leq 0 \\ 0 & \text{ow} \end{cases}$$

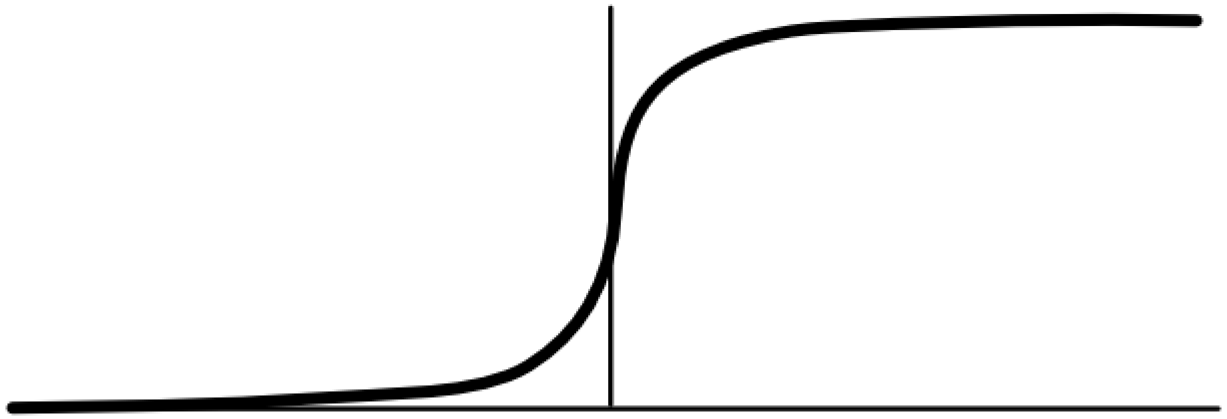
Not a useful function to differentiate

Processing math: 7%

Processing math: 7%

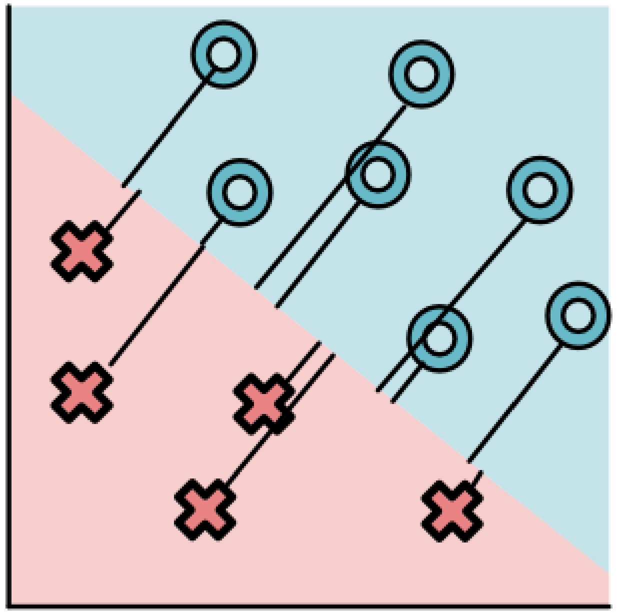
Altenative Function: Sigmoid

Used to determine the loss function



Processing math: 7%

Processing math: 7%



Processing math: 7%

Processing math: 7%

Soft (arg)max?

Would be nice to have a version that with a useful derivative

\text{sigmoid}(x) = \text{softmax} \{0, x\}

Useful soft version of argmax.

Processing math: 7%

Processing math: 7%

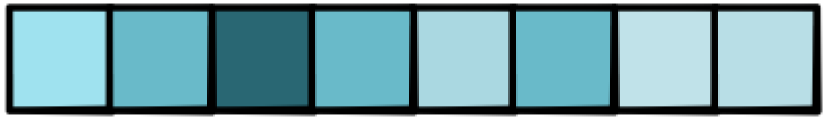
Max, Argmax, Softmax

Processing math: 7%

Processing math: 7%

Challenge

How do we generalize sigmoid to multiple outputs?



Processing math: 7%

Processing math: 7%

Max reduction

- Max is a binary associative operator
- $\max(a, b)$ returns max value
- Generalized $\text{ReLU}(a) = \max(a, 0)$

Max Pooling

- Common to apply pooling with max
- Sets pooled value to "most active" in block
- Forward code is easy to implement

Processing math: 7%

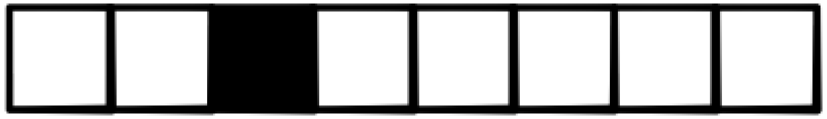
Max Backward

- Unlike sum, max throws away other values
- Only top value gets used
- Backward needs to know this.

Processing math: 7%

Argmax

- Function that returns `argmax`, one-hot
- Generalizes step



Processing math: 7%

Processing math: 7%

Max Backward

- First compute `argmax`
- Only send gradient to `argmax` gradinput
- Everything else is 0

Processing math: 7%

Ties

- What if there are two or more argmax's?
- Max is non-differentiable, like `ReLU(0)`.
- Short answer: Ignore, pick one

Processing math: 7%

Processing math: 7%

HW

- When writing tests for max, ties will break finite-differences
- Suggestion: perturb your input by adding a small amount of random noise.

Processing math: 7%

Processing math: 7%

Soft argmax?

- Need a soft version of argmax.
- Generalizes sigmoid for our new loss function
- Standard name -> softmax

Processing math: 7%

Softmax

\text{softmax}(\textbf{x}) = \frac{\exp \textbf{x}}{\sum_i \exp x_i}

Processing math: 7%

Processing math: 7%

Sigmoid is Softmax

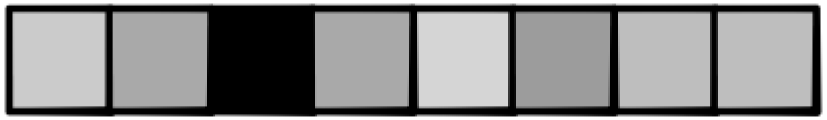
$$\text{softmax}([0, x])[1] = \frac{\exp x}{\exp x + \exp 0} = \sigma(x)$$

Processing math: 7%

Processing math: 7%

Softmax

Softmax



Processing math: 7%

Processing math: 7%

Review

- ReLU -> Max
- Step -> Argmax
- Sigmoid -> Softmax

Processing math: 7%

Processing math: 7%

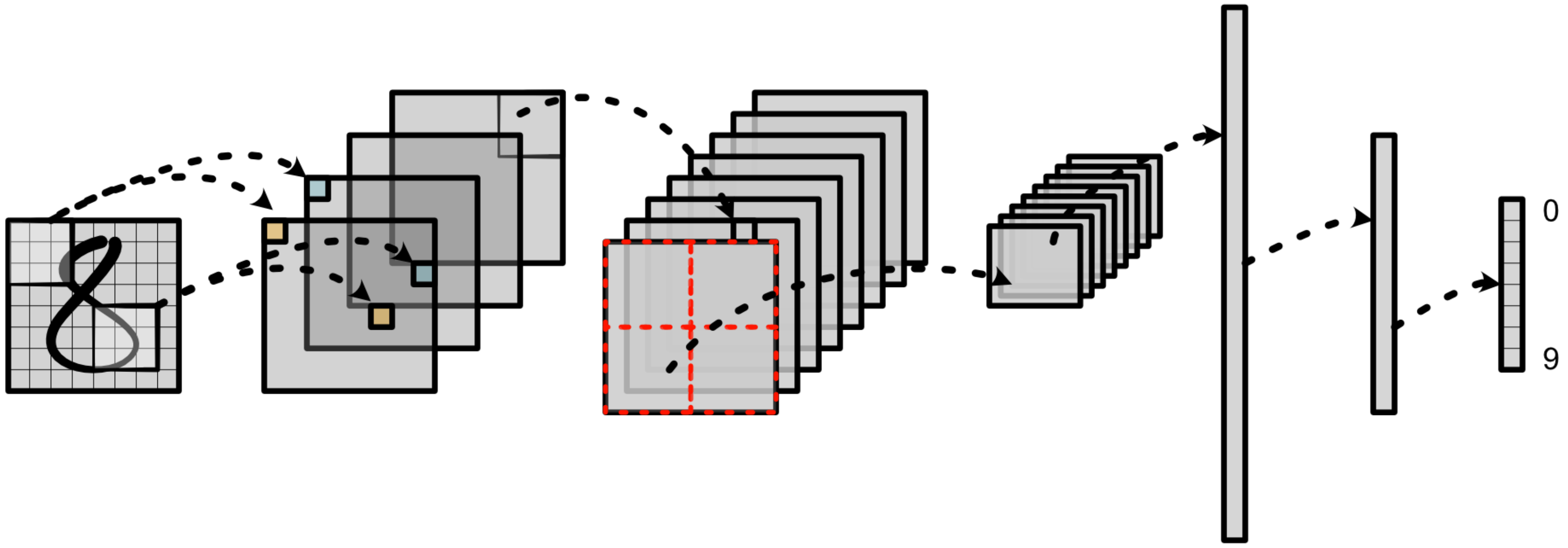
Softmax

Processing math: 7%

Processing math: 7%

Network

Network



Processing math: 7%

Processing math: 7%

Softmax Layer

- Produces a probability distribution over outputs (Sum to 1)
- Derivative similar to sigmoid
- Lots of interesting practical properties

Processing math: 7%

Softmax in Context

- Not a map!
- Gradient spreads out from one point to all.

Processing math: 7%

Processing math: 7%

Softmax

- (Colab) [https://colab.research.google.com/drive/1EB7MI_3gzAR1g

Processing math: 7%

Processing math: 7%

Soft Gates

Processing math: 7%

Processing math: 7%

New Methods

- Sigmoid and softmax produce distributions
- Can be used to "control" information flow

Processing math: 7%

Example

Returns a combination of x and y f(x, y, r) = x * \sigma(r) + y * (1 - \sigma(r))

Processing math: 7%

Gradient is controlled

$$\begin{aligned} f'_x(x,y,r) &= \sigma(r) \\ f'_y(x,y,r) &= (1-\sigma(r)) \\ f'_r(x,y,r) &= (x-y)\sigma'(r) \end{aligned}$$

Processing math: 7%

Processing math: 7%

Neural Network Gates

Learn which one of the previous layers is most useful.

$$r = NN_1$$
$$x = NN_2$$
$$y = NN_3$$

Processing math: 7%

Processing math: 7%

Gradient Flow

- Layers that are used get more updates
- Gradient signals which aspect was important
- Can have extra layers

Processing math: 7%

Selecting Choices

- Gating gives us a binary choice
- What if we want to select between many elements?
- Softmax!

Processing math: 7%

Softmax Gating

Combines many elements of X based on R

$f(X, R) = X \times softmax(R)$

Processing math: 7%

Softmax Gating

- Brand name: Attention

Processing math: 7%

Processing math: 7%

Example: Translation

- Show example

Processing math: 7%

Processing math: 7%

Example: GPT-3

- Show example

Processing math: 7%

Processing math: 7%

QA

Processing math: 7%

Processing math: 7%