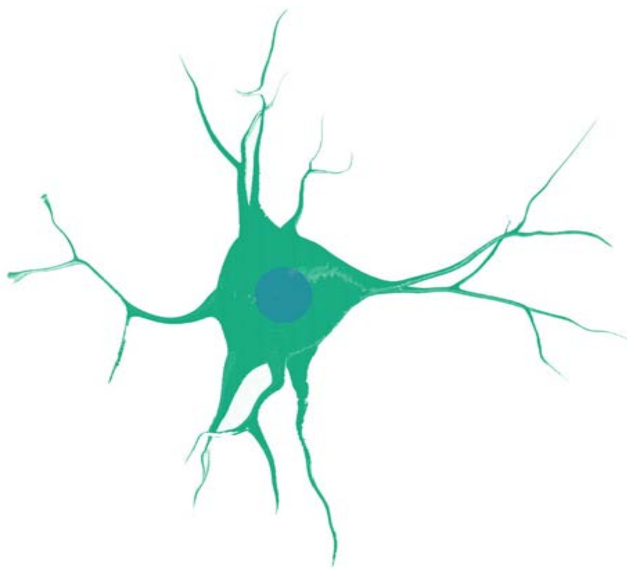
Neural Networks

Cynthia Rudin

Duke Machine Learning

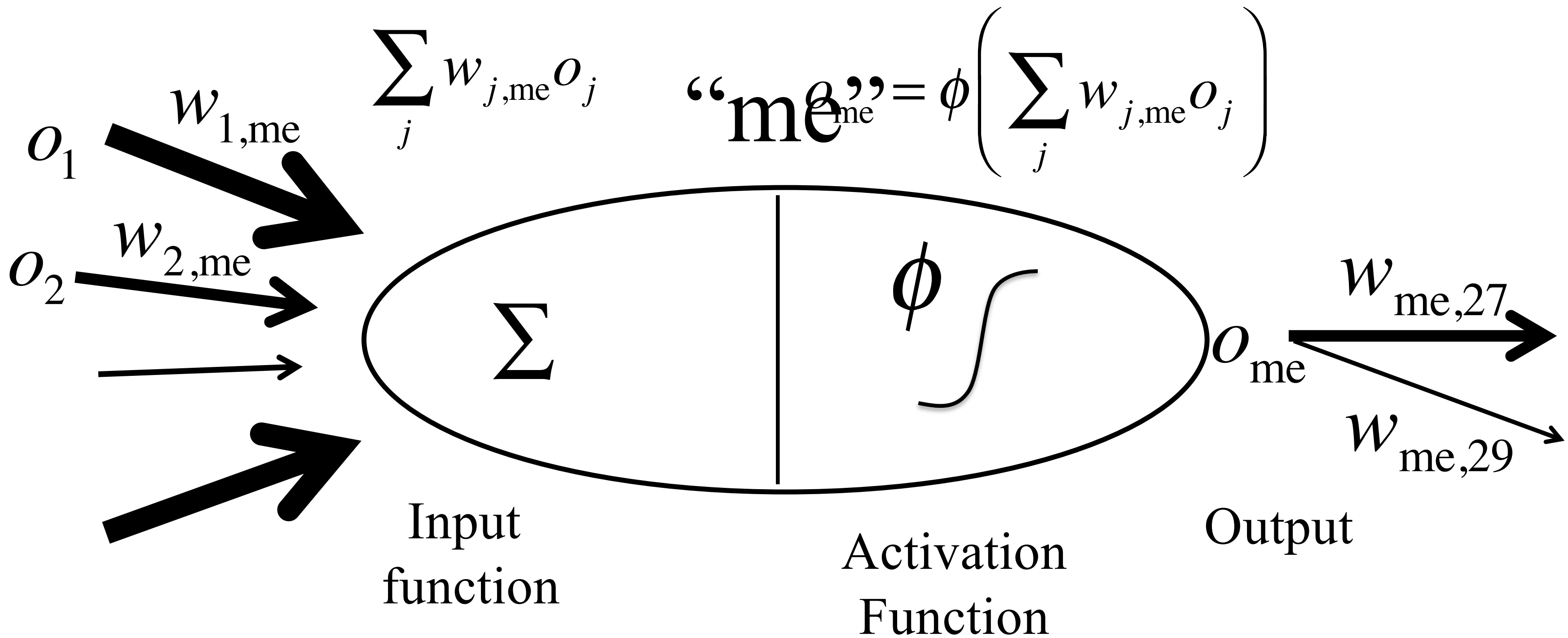
Neurons

- 10^{11} neurons in a brain, 10^{14} synapses (connections).
- Signals are electrical potential spikes that travel through the network.

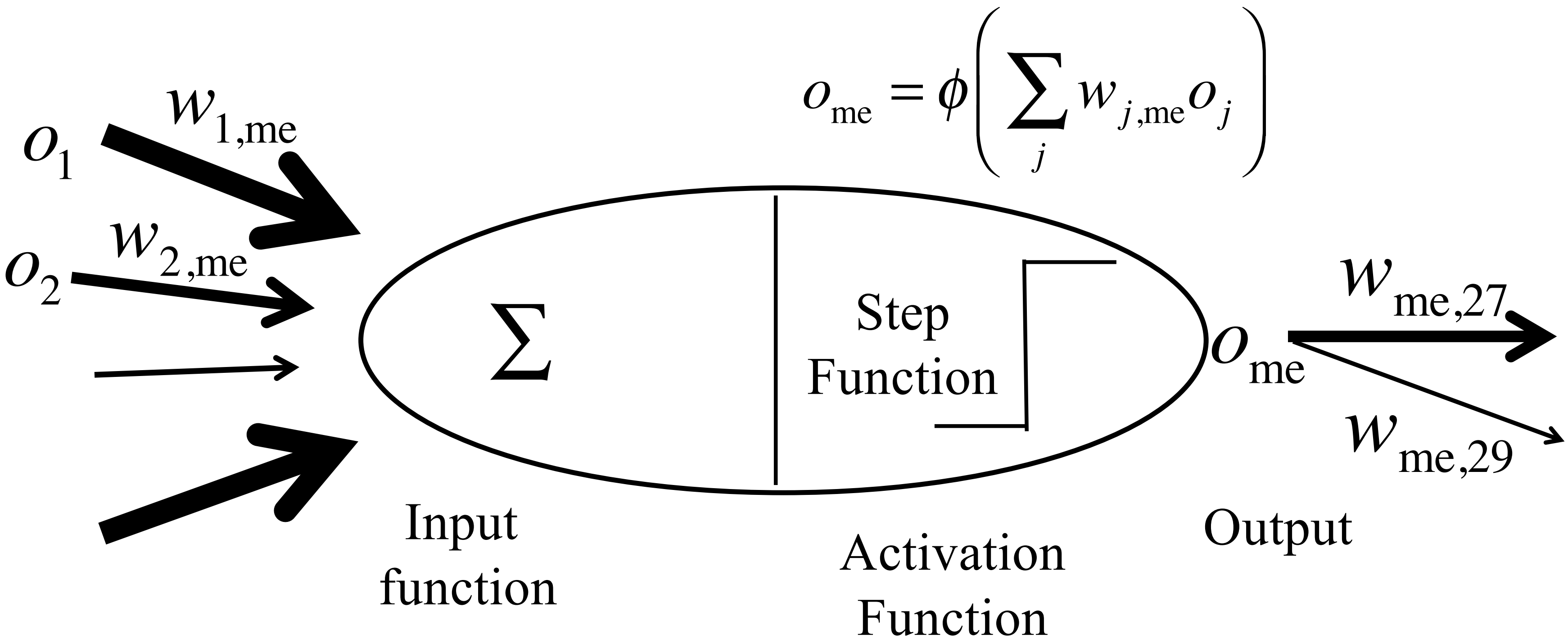


(Credit: Adapted from Russell and Norvig)

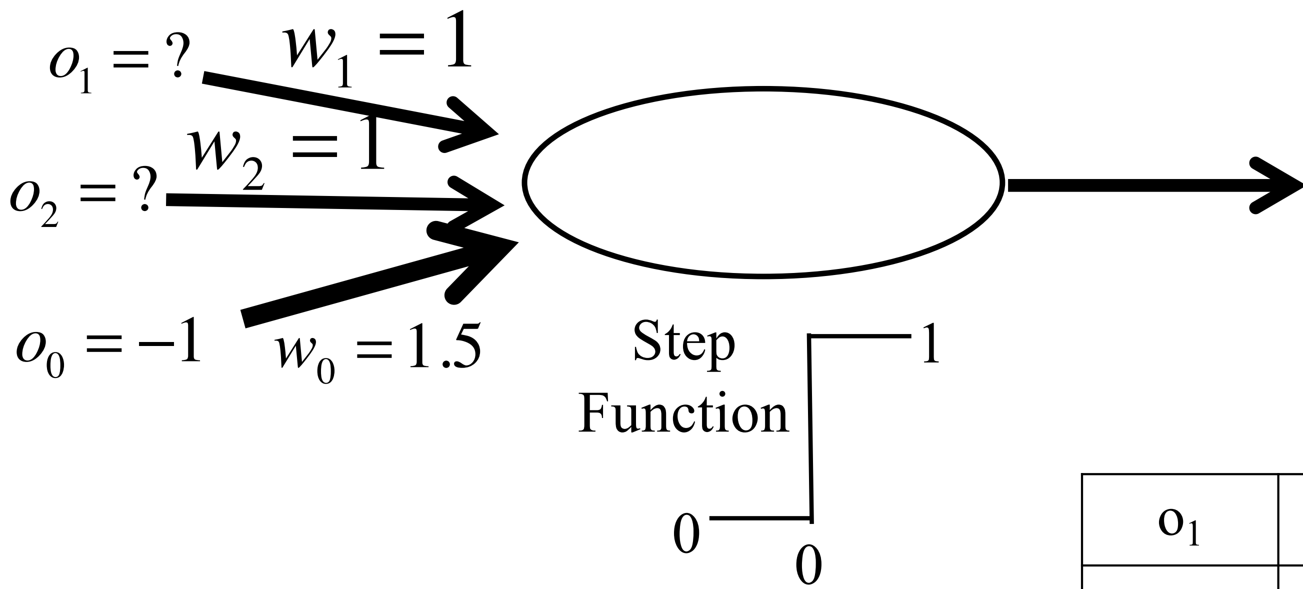
McCulloch-Pitts "Neuron"



McCulloch-Pitts “Neuron”

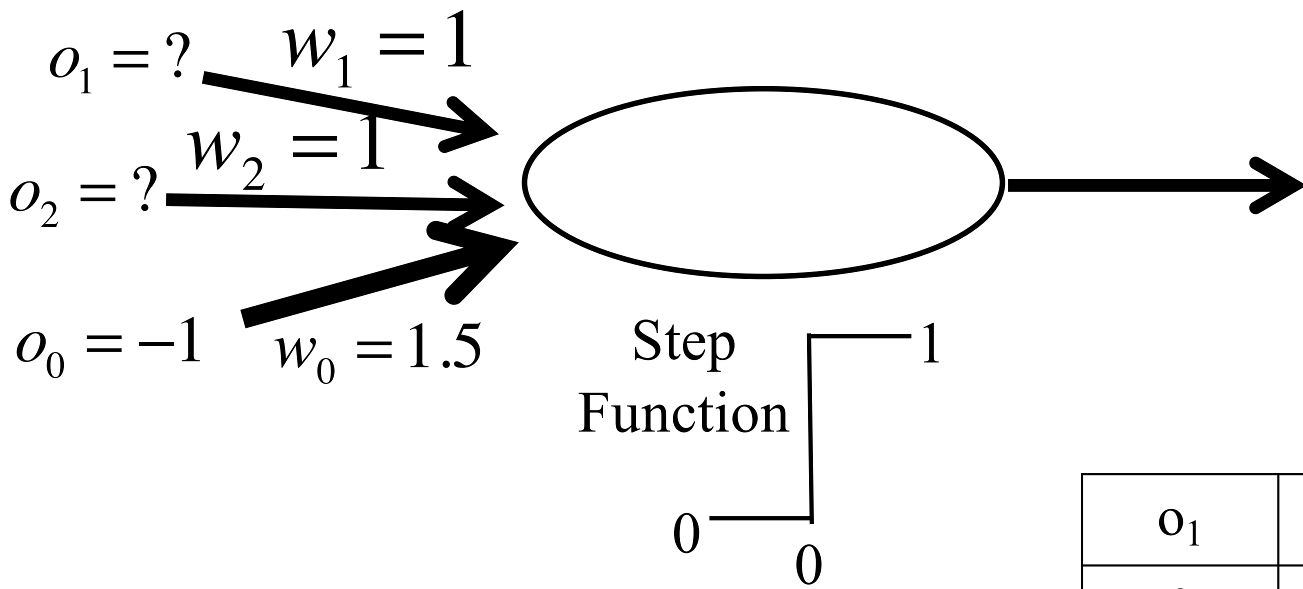


McCulloch-Pitts “Neuron”



o_1	o_2	output

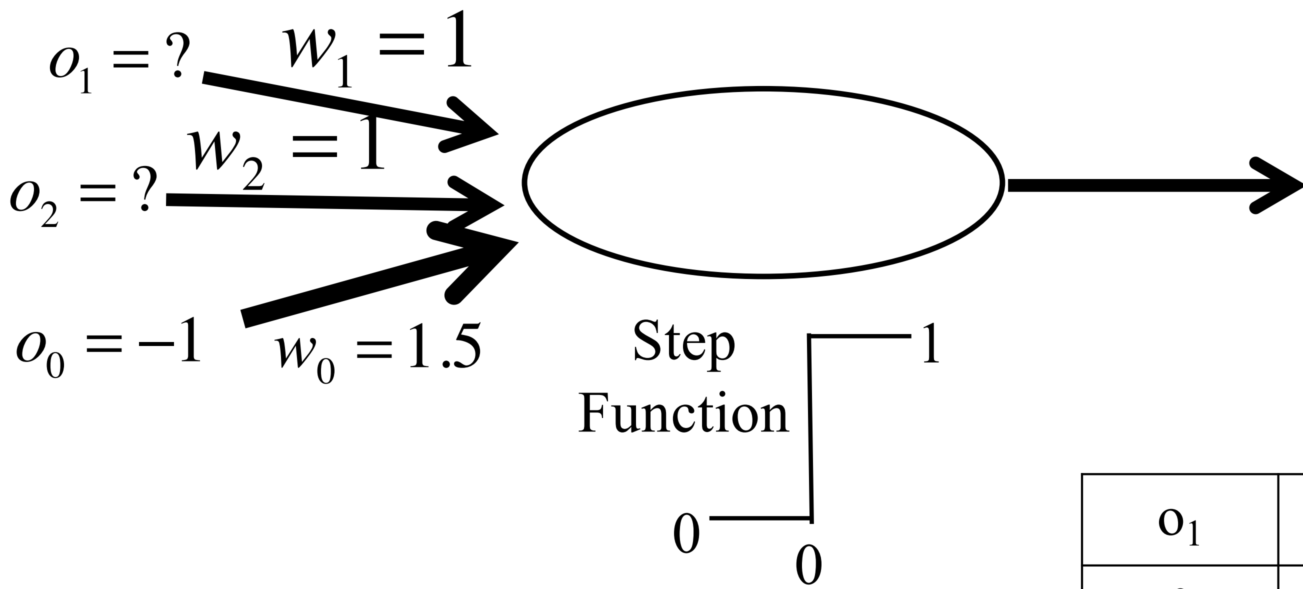
McCulloch-Pitts “Neuron”



o_1	o_2	output
0	0	

0+0-1.5 = -1.5

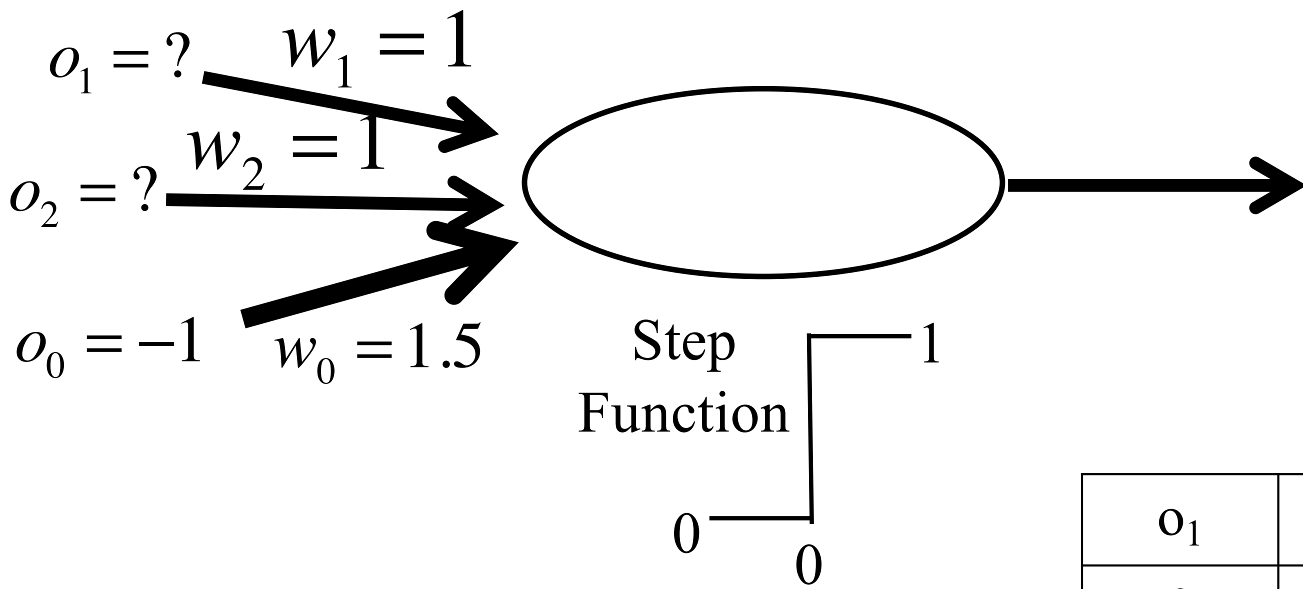
McCulloch-Pitts “Neuron”



o_1	o_2	output
0	0	0

0+0-1.5 = -1.5

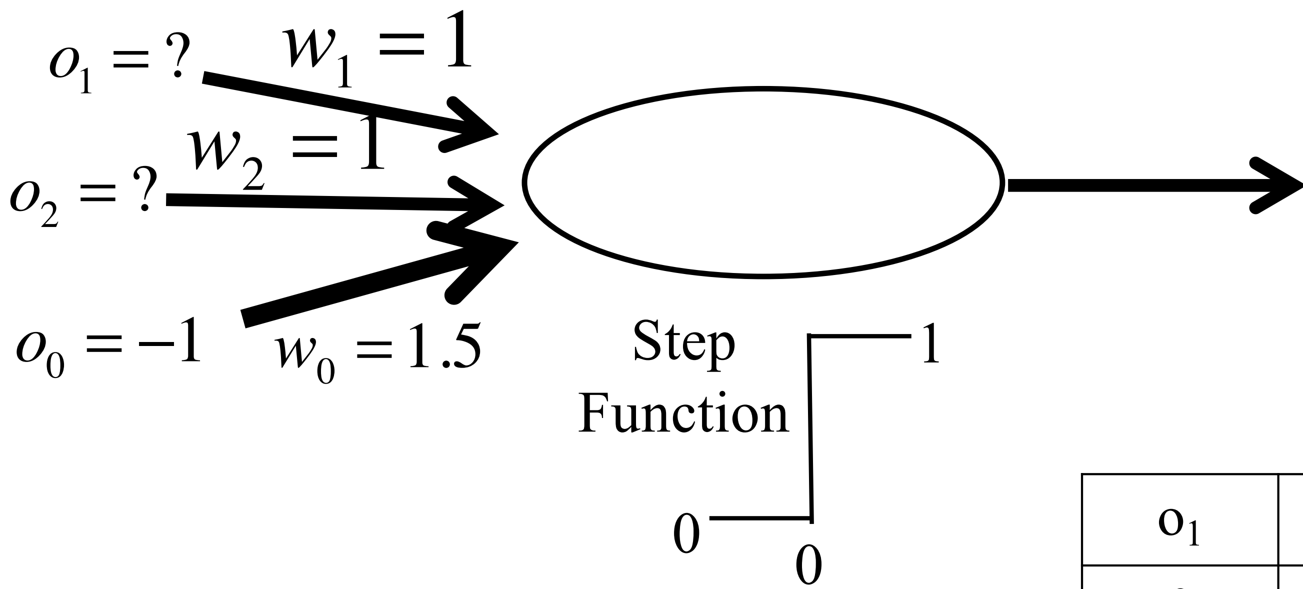
McCulloch-Pitts “Neuron”



o_1	o_2	output
0	0	0
1	0	0

1+0-1.5 = -0.5

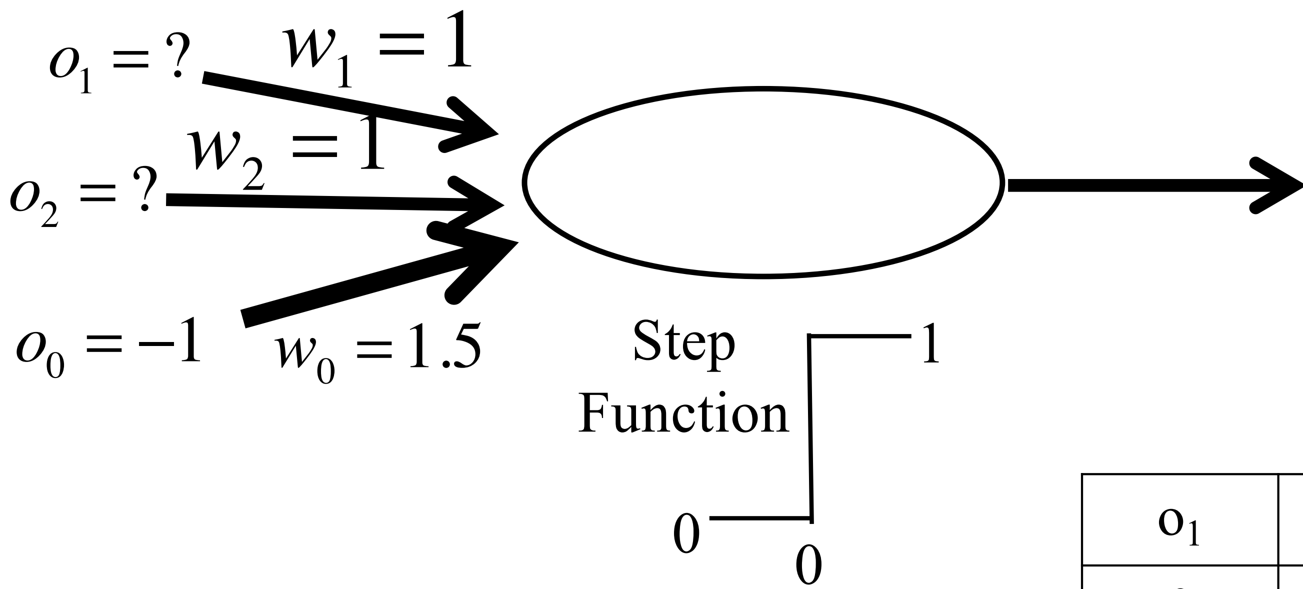
McCulloch-Pitts "Neuron"



o_1	o_2	output
0	0	0
1	0	0
0	1	0

0+1-1.5 = -0.5

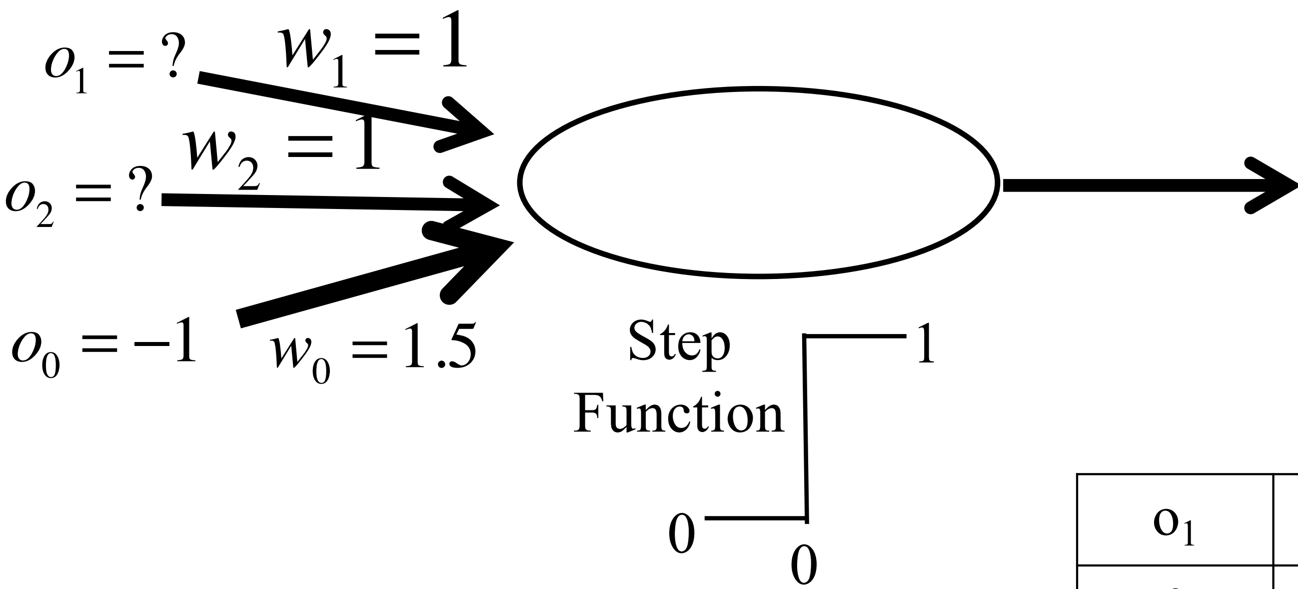
McCulloch-Pitts "Neuron"



o_1	o_2	output
0	0	0
1	0	0
0	1	0
1	1	

1+1-1.5 = 0.5

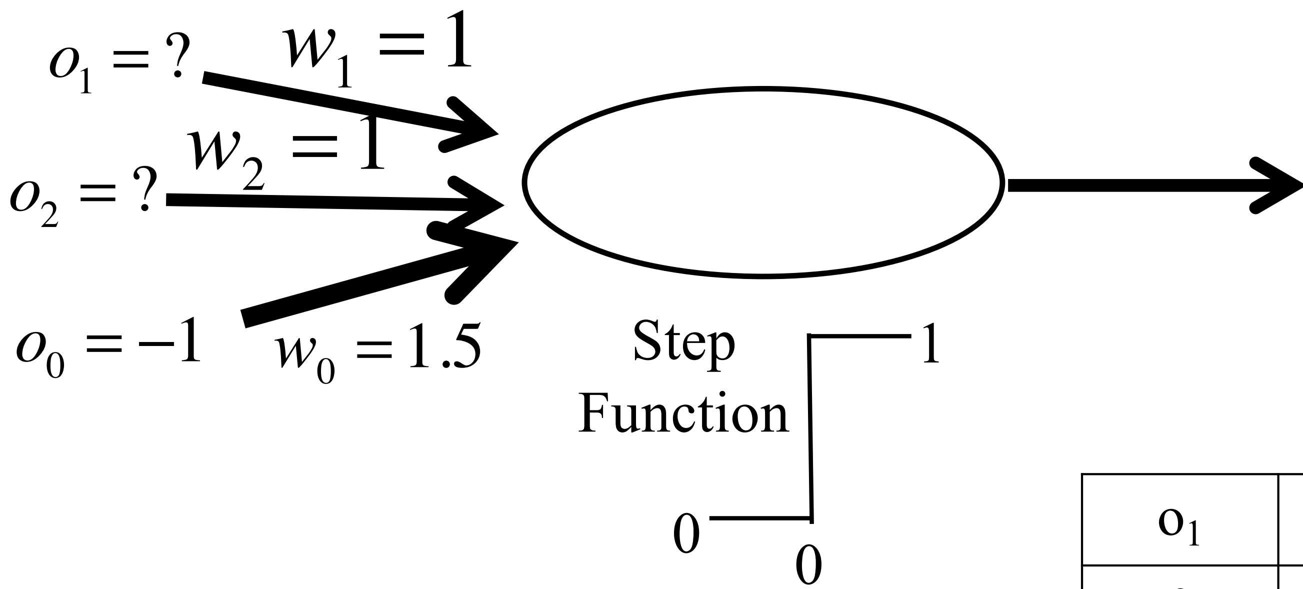
McCulloch-Pitts "Neuron"



o_1	o_2	output
0	0	0
1	0	0
0	1	0
1	1	1

1+1-1.5 = 0.5

McCulloch-Pitts “Neuron”



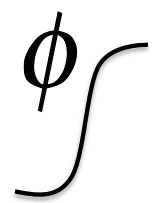
This neuron computes the function “and.”

There are “or” and “not” neurons too.

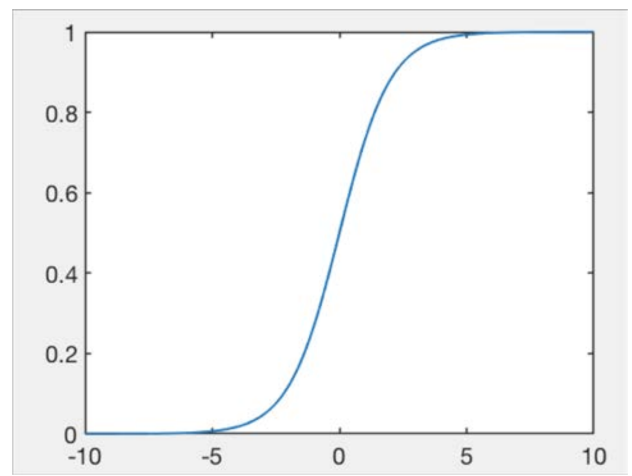
o_1	o_2	output
0	0	0
1	0	0
0	1	0
1	1	1

McCulloch-Pitts “Neuron”

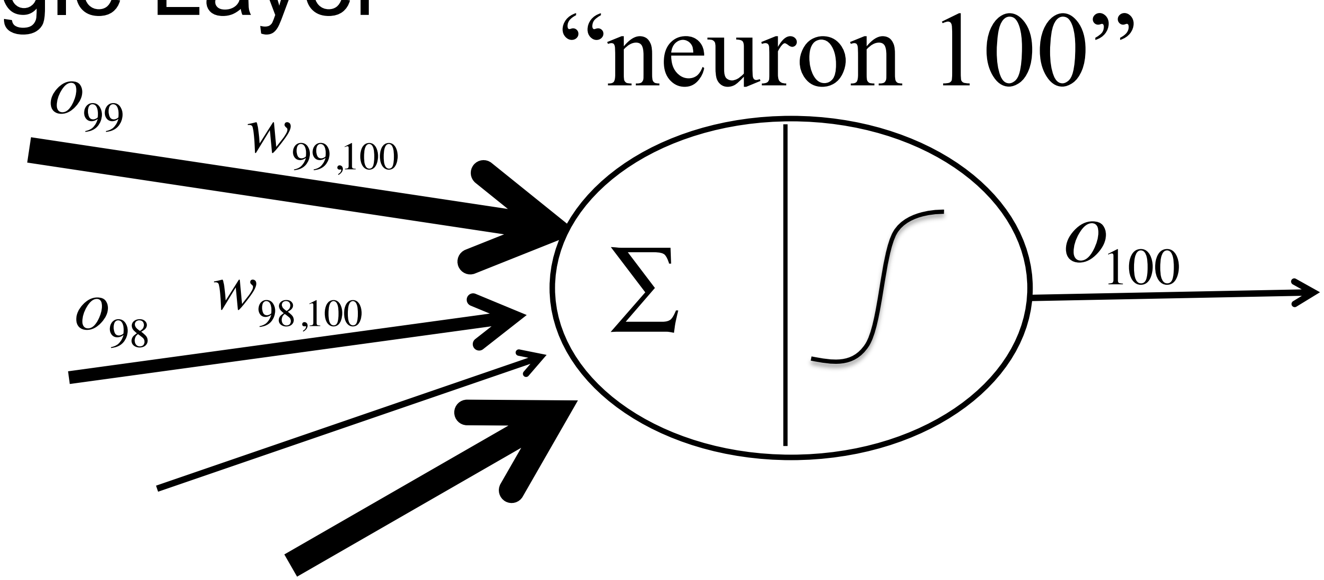
$$\phi\left(\sum_j w_{j,\mathrm{me}} a_j\right) = 1/(1+e^{-x}) \quad \text{“Sigmoid”}$$



Activation Function



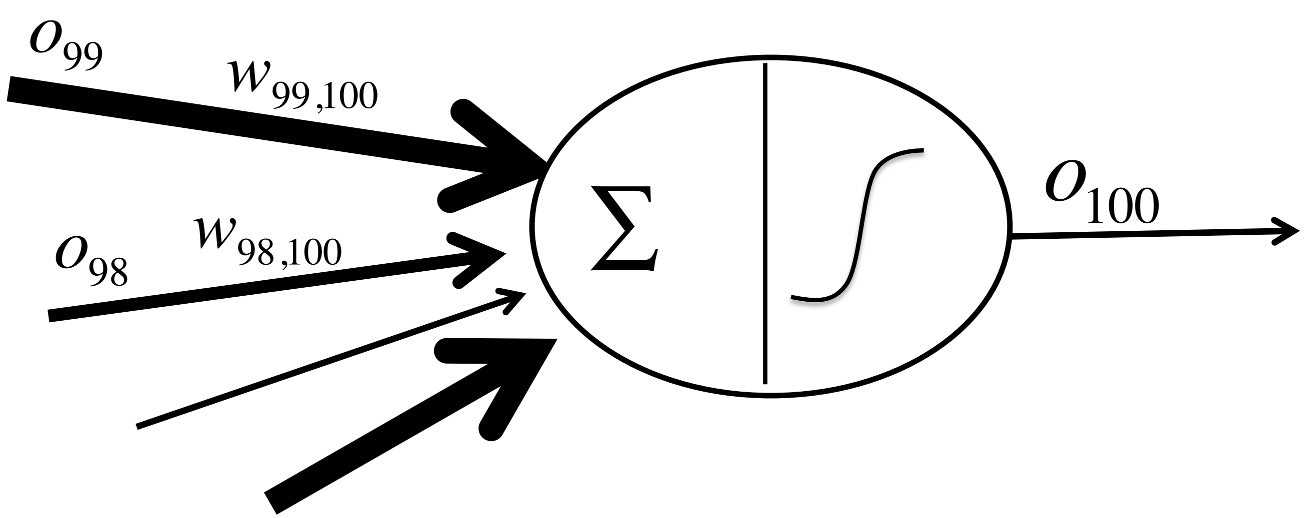
Single Layer



Error on one observation

$$E = \frac{1}{2}(y - o_{100})^2$$

Single Layer



Error on one observation

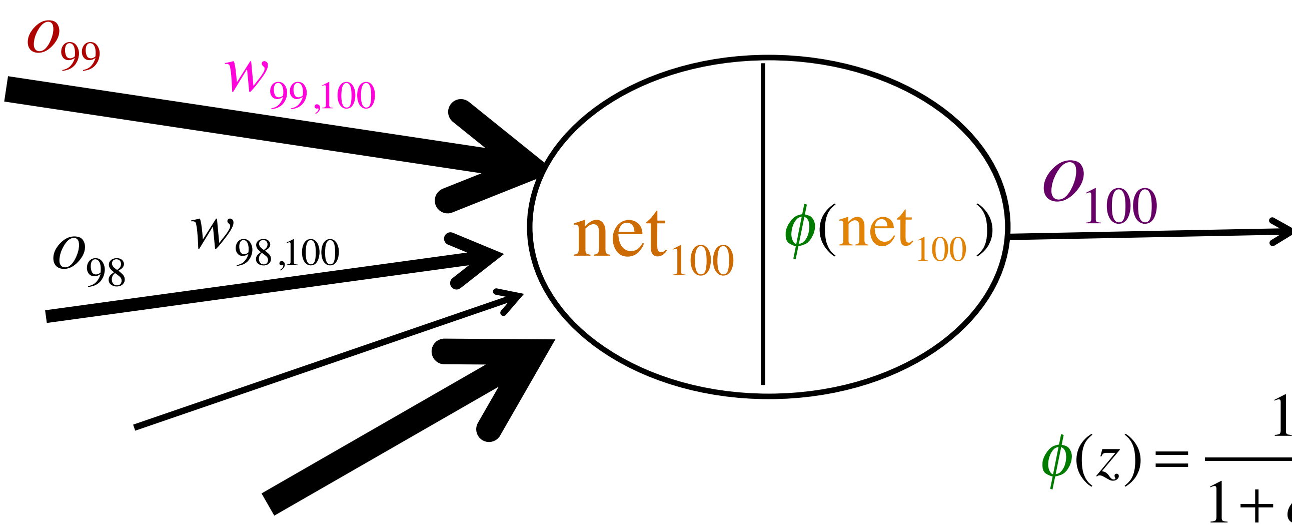
$$E = \frac{1}{2}(y - o_{100})^2$$

- In a brain, the synapses strengthen and weaken in order to learn.
- Say the same thing happens here.
- How should we set the weights in order to learn (reduce the error)?
- Minimize E with respect to the weights.

Backpropagation

- An algorithm that trains the weights of a neural network
- Requires us to propagate information backwards through the network, then forwards, then backwards, then forwards, etc.
- Propagate backwards = chain rule from calculus.

Single Layer



Error on one observation

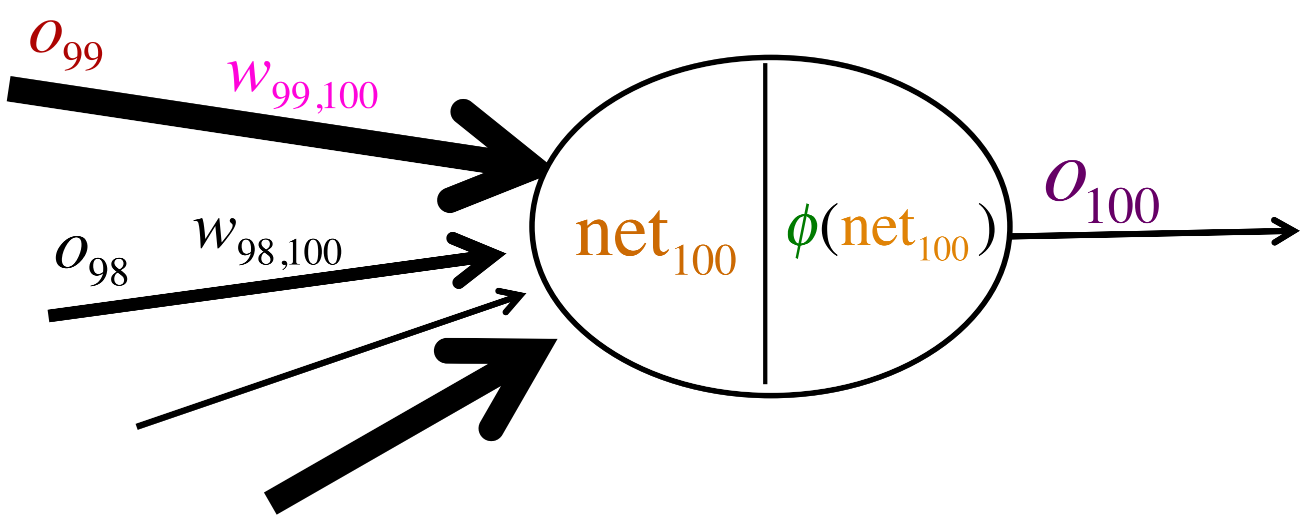
$$E = \frac{1}{2}(y - o_{100})^2$$

$$\phi(z) = \frac{1}{1+e^{-z}}$$

$$\phi'(z) = \frac{d\phi(z)}{dz} = \phi(z)(1-\phi(z))$$

$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \text{net}_{100}} \frac{d\ \text{net}_{100}}{dw_{99,100}}$$

Single Layer



Error on one observation

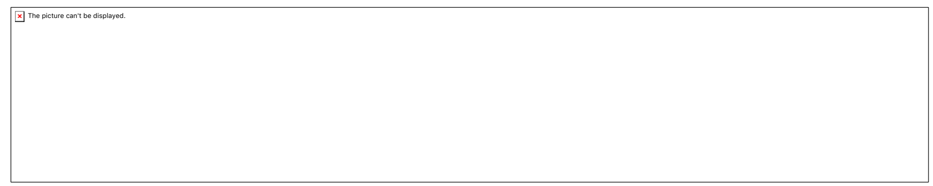
$$E = \frac{1}{2}(y - o_{100})^2$$

$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \text{net}_{100}} \frac{d\ \text{net}_{100}}{dw_{99,100}}$$

Single Layer

Error on one observation

$$E = \frac{1}{2}(y - o_{100})^2$$



$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \mathrm{net}_{100}} \frac{d\ \mathrm{net}_{100}}{dw_{99,100}}$$

$$-(y - o_{100})$$

Single Layer

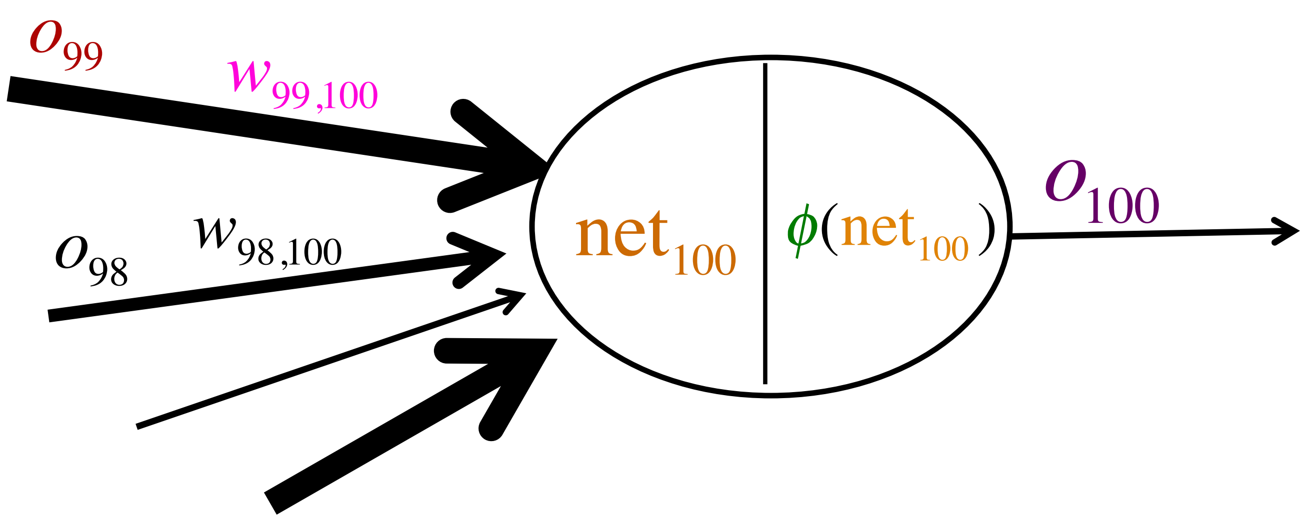
Error on one observation

$$E = \frac{1}{2}(y - o_{100})^2$$

$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \text{net}_{100}} \frac{d\ \text{net}_{100}}{dw_{99,100}}$$

$$-(y - o_{100})$$

Single Layer



Error on one observation

$$E = \frac{1}{2}(y - o_{100})^2$$

$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \text{net}_{100}} \frac{d\ \text{net}_{100}}{dw_{99,100}}$$

$-(y - o_{100})$

Single Layer

$$\phi(\text{net}_{100}) \qquad o_{100}$$

$$\frac{d o_{100}}{d\ \text{net}_{100}} = \frac{d\phi(\text{net}_{100})}{d\ \text{net}_{100}} = \phi'(\text{net}_{100}) = \phi(\text{net}_{100})(1-\phi(\text{net}_{100})) = o_{100}(1-o_{100})$$

$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \text{net}_{100}} \frac{d\ \text{net}_{100}}{dw_{99,100}}$$

$$-(y-o_{100})$$

Single Layer

$$\phi(\text{net}_{100}) \quad o_{100}$$

$$\frac{do_{100}}{d\ \text{net}_{100}} = \frac{d\phi(\text{net}_{100})}{d\ \text{net}_{100}} = \phi'(\text{net}_{100}) = \phi(\text{net}_{100})(1-\phi(\text{net}_{100})) = o_{100}(1-o_{100})$$

$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \text{net}_{100}} \frac{d\ \text{net}_{100}}{dw_{99,100}}$$

$$-(y-o_{100})$$

$$o_{100}(1-o_{100})$$

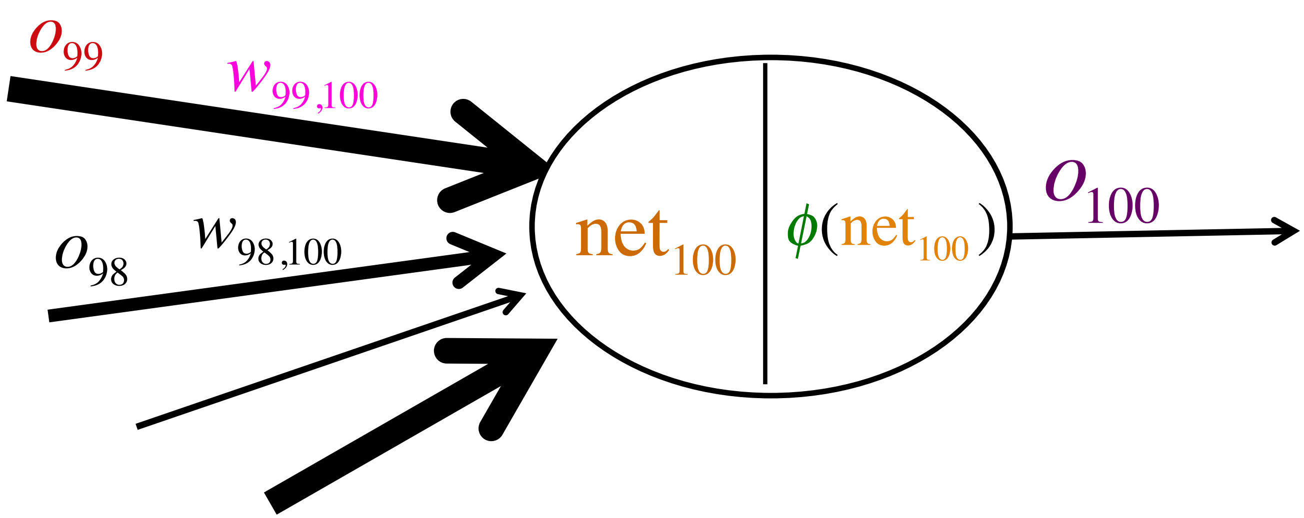
Single Layer

$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \text{net}_{100}} \frac{d\ \text{net}_{100}}{dw_{99,100}}$$

$-(y - o_{100})$

$o_{100}(1 - o_{100})$

Single Layer



Error on one observation

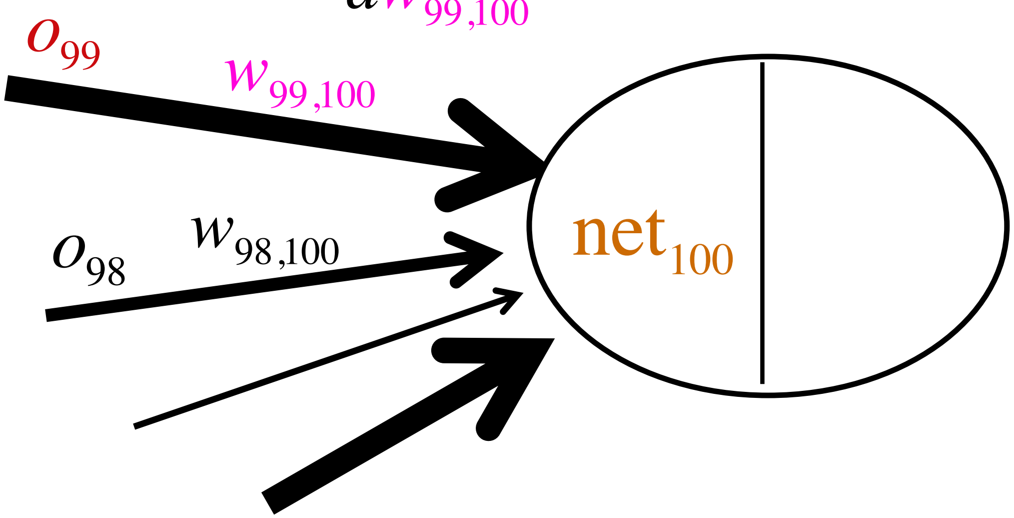
$$E = \frac{1}{2}(y - o_{100})^2$$

$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \text{net}_{100}} \frac{d\ \text{net}_{100}}{dw_{99,100}}$$

$-(y - o_{100})$

$o_{100}(1 - o_{100})$

$$\frac{d\ \text{net}_{100}}{dw_{99,100}} = \frac{d\ (w_{99,100}o_{99} + w_{98,100}o_{98} + w_{97,100}o_{97} + \ldots)}{dw_{99,100}} = o_{99}$$



$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \text{net}_{100}} \frac{d\ \text{net}_{100}}{dw_{99,100}}$$

o_{99}

$-(y - o_{100})$

$o_{100}(1 - o_{100})$

$$\frac{dE}{dw_{99,100}} = \frac{dE}{do_{100}} \frac{do_{100}}{d\ \mathrm{net}_{100}} \frac{d\ \mathrm{net}_{100}}{dw_{99,100}}$$

o_{99}

$-(y - o_{100})$

$o_{100}(1 - o_{100})$

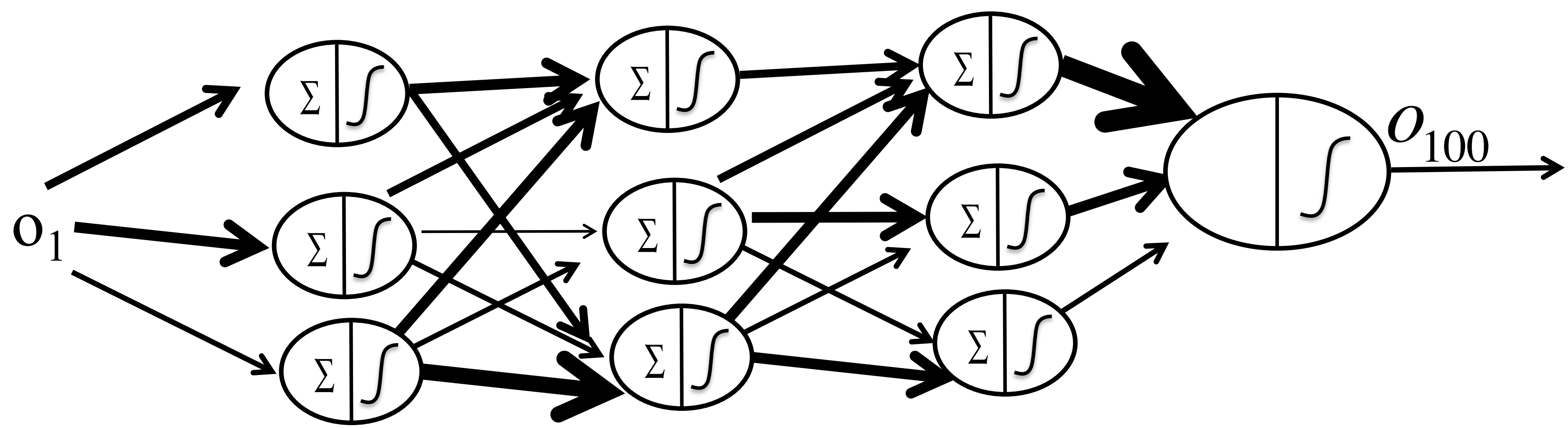
We will need this later – it depends only on node 100

$$\frac{dE}{dw_{99,100}} = \overbrace{\frac{dE}{do_{100}} \frac{do_{100}}{d\ \mathrm{net}_{100}}}^{\delta_{100}} \frac{d\ \mathrm{net}_{100}}{dw_{99,100}}$$

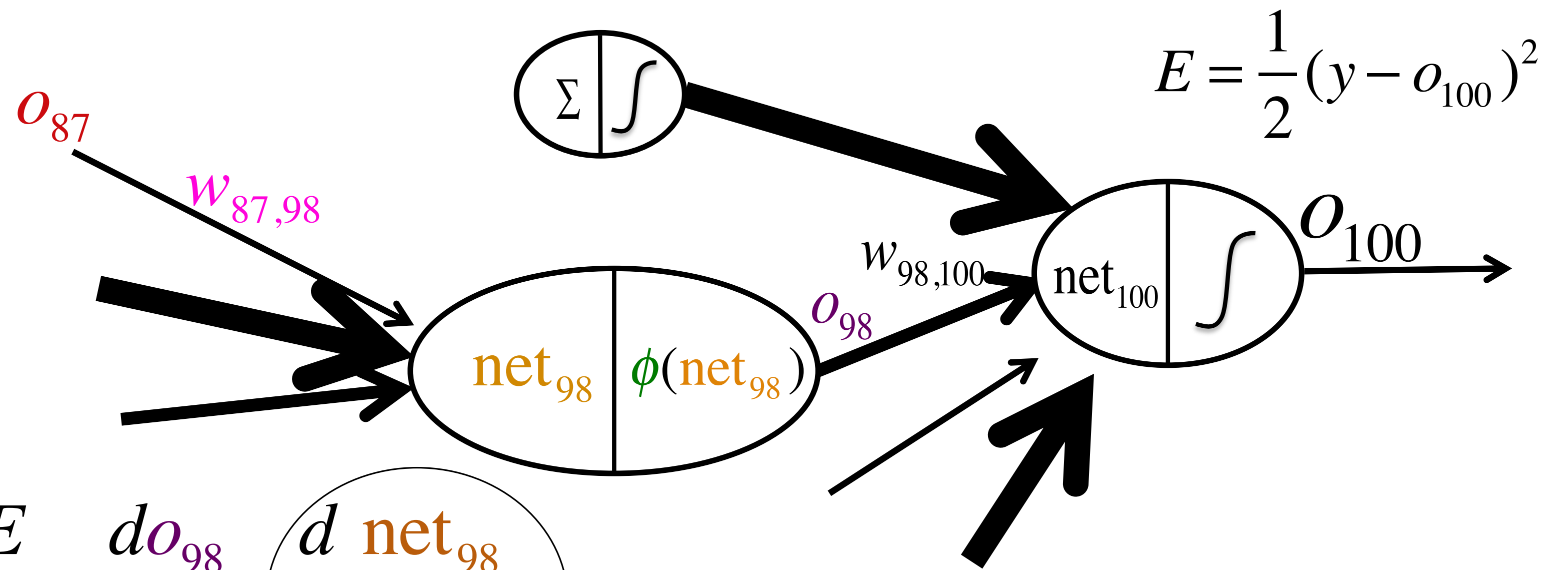
$$= (-(y - o_{100}))o_{100}(1 - o_{100})\, o_{99}$$

Backpropagation

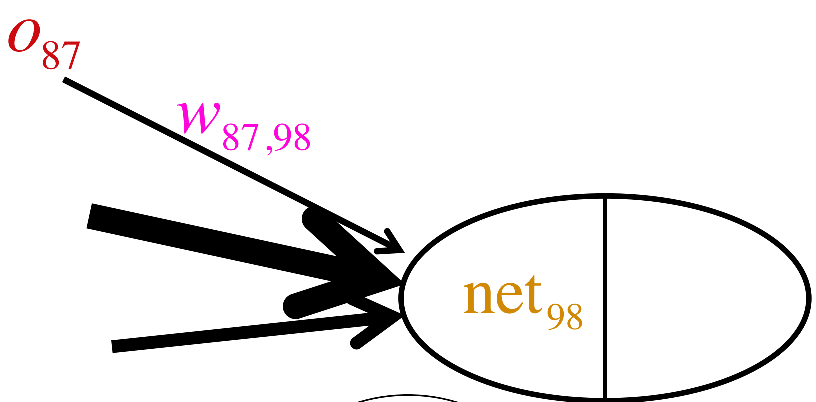
- Go one layer deeper.



o1
o100

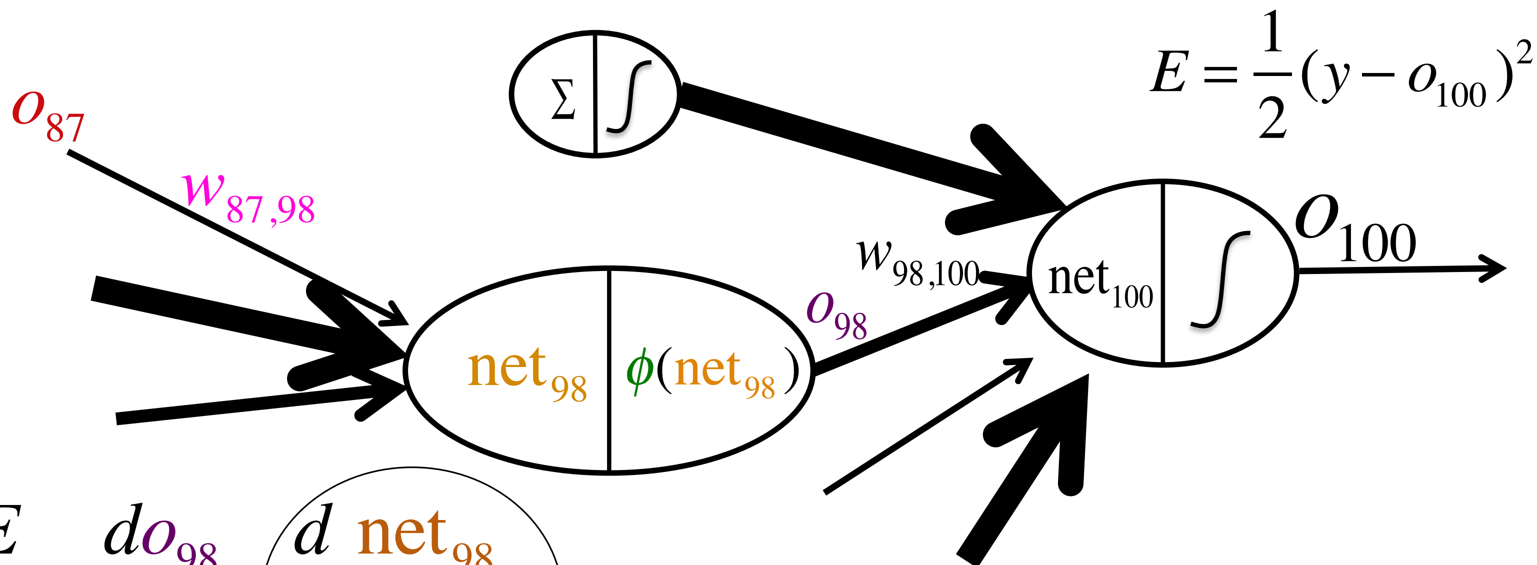


$$\frac{dE}{dw_{87,98}} = \frac{dE}{do_{98}} \frac{do_{98}}{d\ \mathrm{net}_{98}} \frac{d\ \mathrm{net}_{98}}{dw_{87,98}}$$



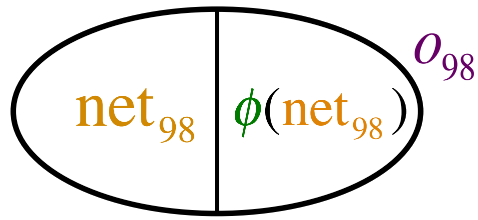
$$\frac{dE}{dw_{87,98}} = \frac{dE}{do_{98}} \frac{do_{98}}{d\ \text{net}_{98}} \frac{d\ \text{net}_{98}}{dw_{87,98}}$$

$$\frac{d\ \text{net}_{98}}{dw_{87,98}} = \frac{d\ (w_{87,98}o_{87} + w_{86,98}o_{86} + w_{85,98}o_{85} + \ldots)}{dw_{87,98}} = o_{87}$$



$$\frac{dE}{dw_{87,98}} = \frac{dE}{do_{98}} \frac{do_{98}}{d\ \text{net}_{98}} \frac{d\ \text{net}_{98}}{dw_{87,98}}$$

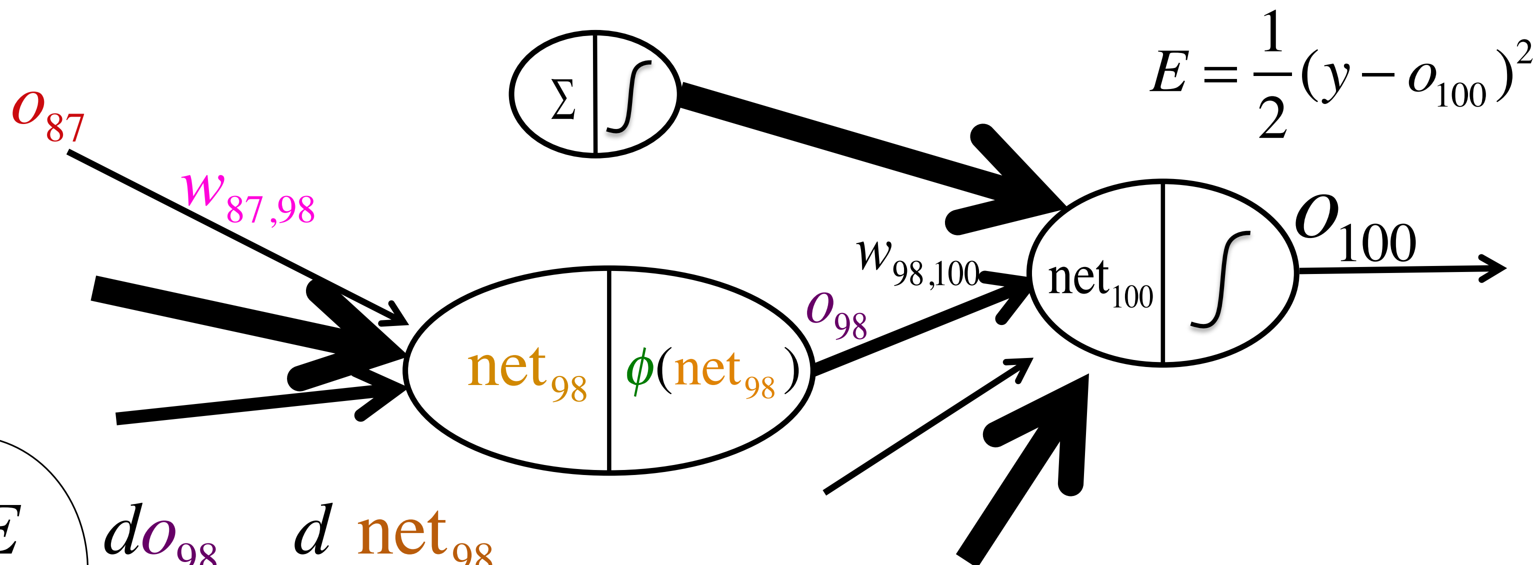
o_{87}



$$\frac{dE}{dw_{87,98}} = \frac{dE}{do_{98}} \frac{do_{98}}{d\ net_{98}} \frac{d\ net_{98}}{dw_{87,98}}$$

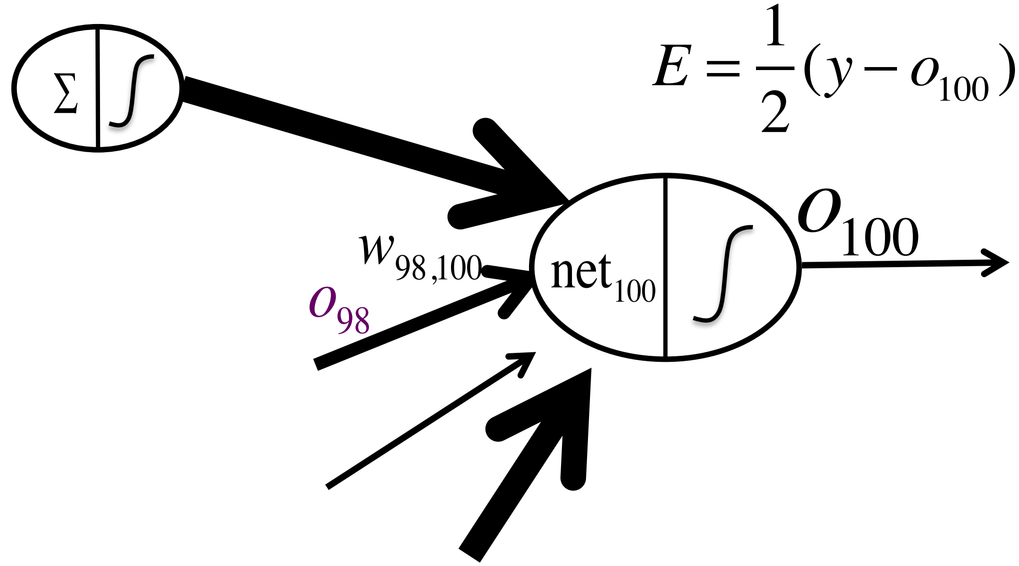
o_{87}

$$\frac{do_{98}}{d\ net_{98}} = \frac{d\phi(net_{98})}{d\ net_{98}} = \phi'(net_{98}) = \phi(net_{98})(1-\phi(net_{98})) = o_{98}(1-o_{98})$$



$$\frac{dE}{dw_{87,98}} = \frac{dE}{do_{98}} \underbrace{\frac{do_{98}}{d\ \mathrm{net}_{98}}}_{o_{98}(1 - o_{98})} \underbrace{\frac{d\ \mathrm{net}_{98}}{dw_{87,98}}}_{o_{87}}$$

$$E = \frac{1}{2}(y - o_{100})^2$$

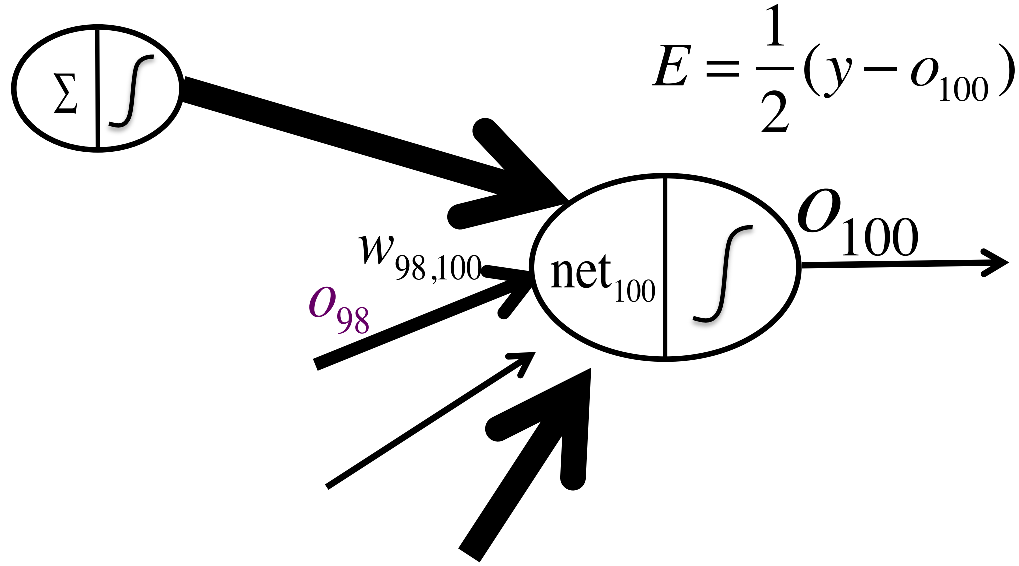


$$\frac{dE}{dw_{87,98}} = \frac{dE}{do_{98}} \frac{do_{98}}{d\ \mathrm{net}_{98}} \frac{d\ \mathrm{net}_{98}}{dw_{87,98}}$$

o_{87}

$o_{98}(1 - o_{98})$

$$E = \frac{1}{2}(y - o_{100})^2$$



$$\frac{dE}{do_{98}} = \frac{dE}{d\text{net}_{100}} \frac{d\text{net}_{100}}{do_{98}}$$

δ_{100}

$$\frac{dE}{dw_{87,98}} = \frac{dE}{do_{98}} \frac{do_{98}}{d\ \text{net}_{98}} \frac{d\ \text{net}_{98}}{dw_{87,98}}$$

o_{87}

$o_{98}(1 - o_{98})$

$$\frac{d\,\text{net}_{100}}{do_{98}} = \frac{d(w_{99,100}o_{99} + w_{98,100}o_{98} + \ldots)}{do_{98}} = w_{98,100}$$

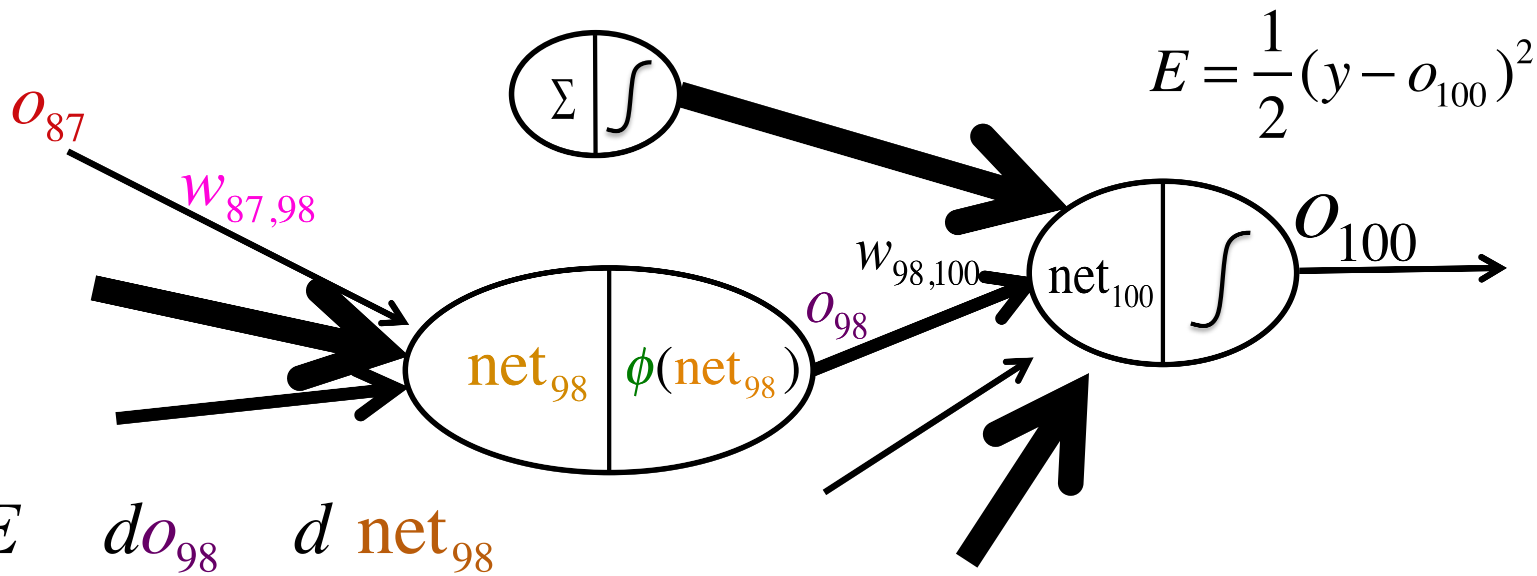
$$\frac{dE}{do_{98}} = \frac{dE}{d\,\text{net}_{100}} \frac{d\,\text{net}_{100}}{do_{98}}$$

δ_{100}

$$\frac{dE}{dw_{87,98}} = \frac{dE}{do_{98}} \frac{do_{98}}{d\ \text{net}_{98}} \frac{d\ \text{net}_{98}}{dw_{87,98}}$$

o_{87}

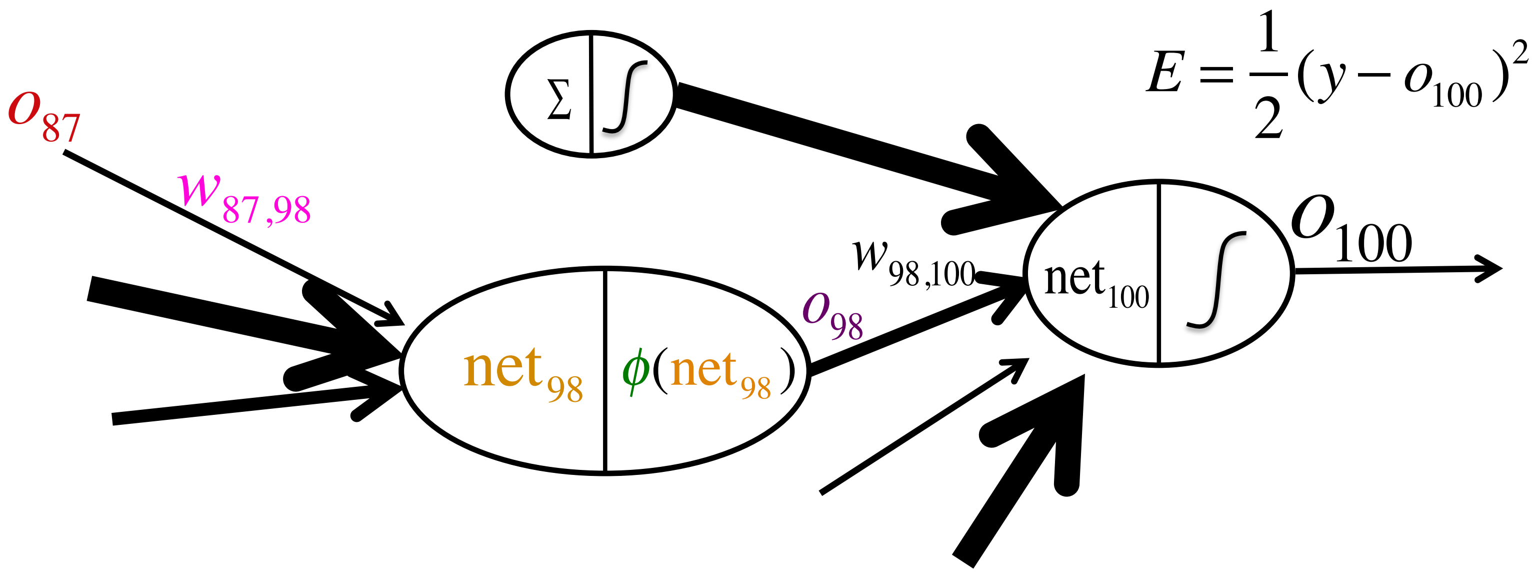
$o_{98}(1 - o_{98})$



$$\frac{dE}{dw_{87,98}} = \frac{dE}{do_{98}} \frac{do_{98}}{d\ \text{net}_{98}} \frac{d\ \text{net}_{98}}{dw_{87,98}}$$

o_{87}

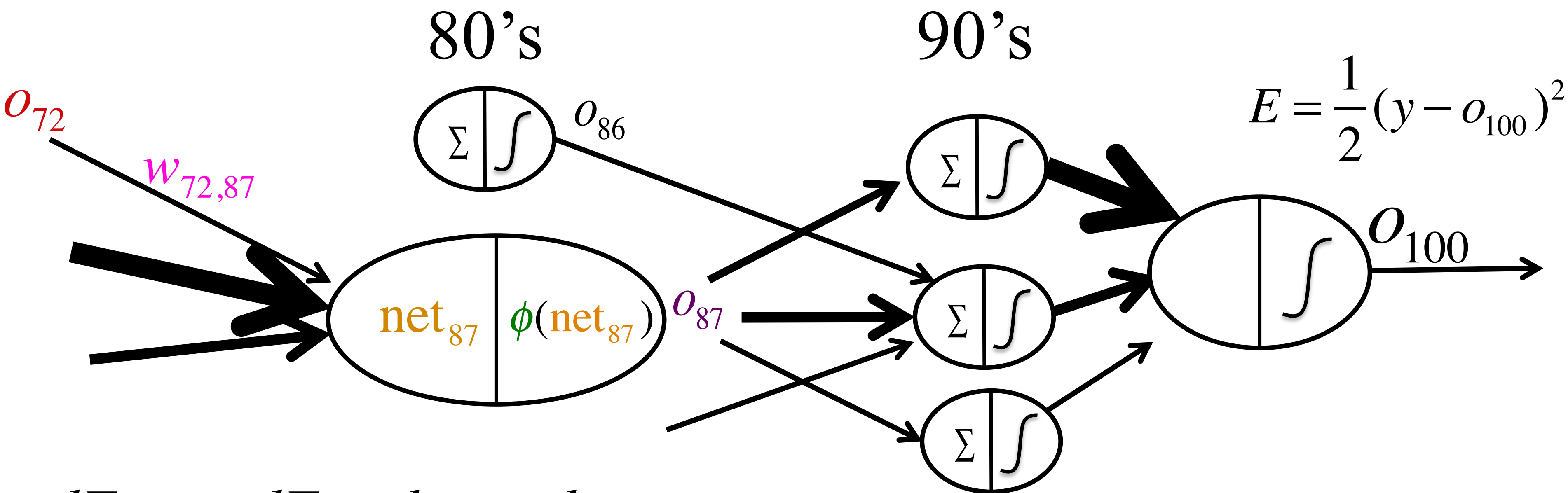
$\delta_{100} w_{98,100}$ $\qquad$ $o_{98}(1 - o_{98})$



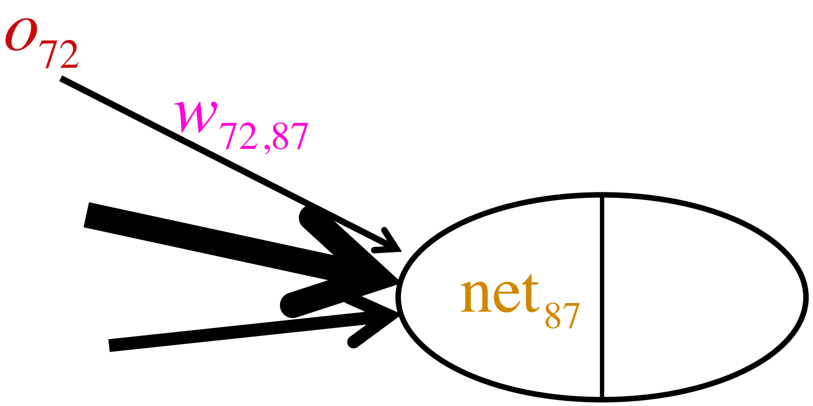
$$\frac{dE}{dw_{87,98}} = \delta_{100} w_{98,100} o_{98} (1 - o_{98}) o_{87}$$

Backpropagation

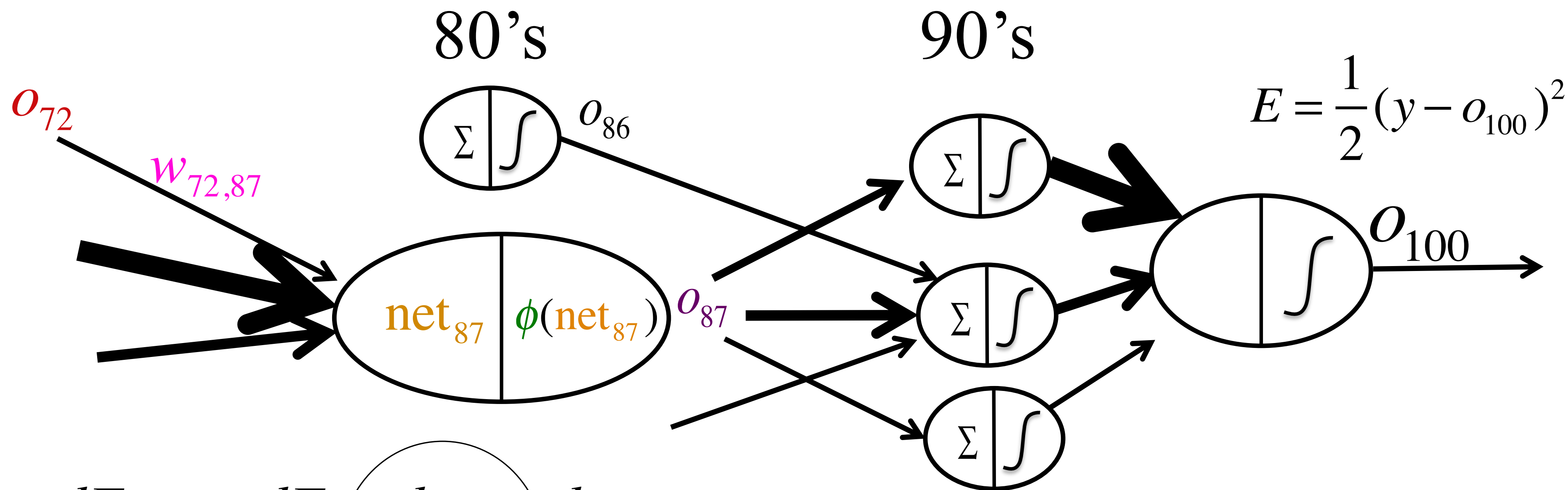
- Go even one layer deeper.
- Third time is a charm.



$$\frac{dE}{dw_{72,87}} = \frac{dE}{do_{87}} \frac{do_{87}}{d\ \text{net}_{87}} \frac{d\ \text{net}_{87}}{dw_{72,87}}$$

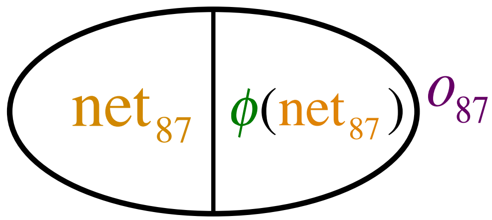


$$\frac{dE}{dw_{72,87}} = \frac{dE}{do_{87}} \frac{do_{87}}{d\ \text{net}_{87}} \underbrace{\frac{d\ \text{net}_{87}}{dw_{72,87}}}_{o_{72}}$$



$$\frac{dE}{dw_{72,87}} = \frac{dE}{do_{87}} \frac{do_{87}}{d\ \text{net}_{87}} \frac{d\ \text{net}_{87}}{dw_{72,87}}$$

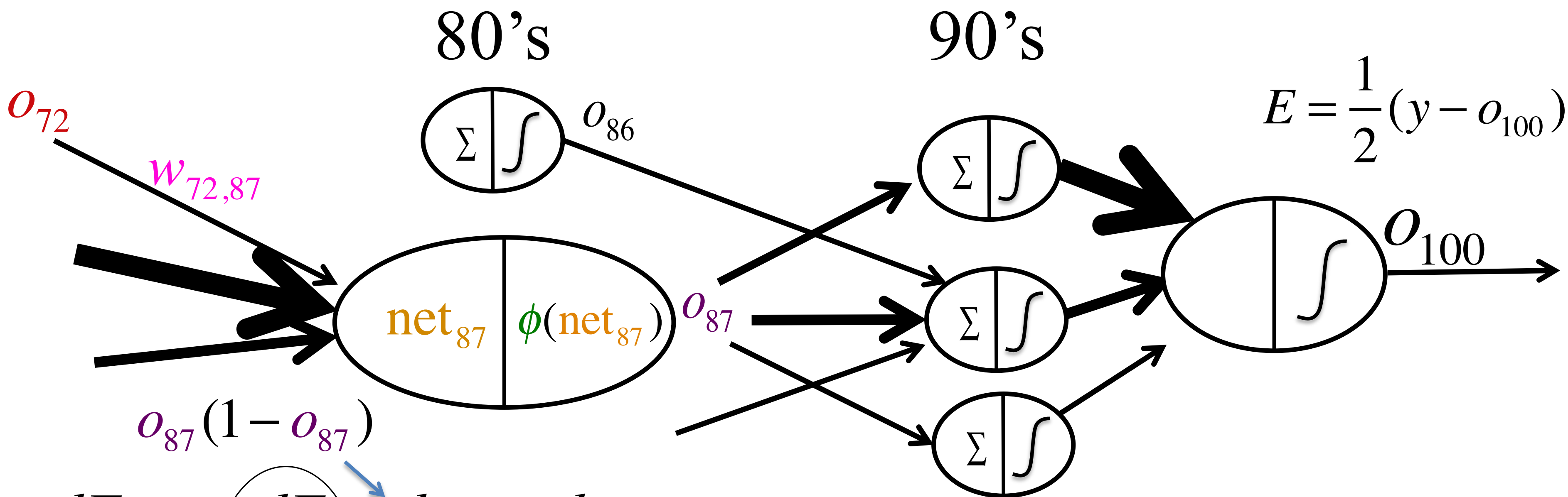
o_{72}



$$\frac{dE}{dw_{72,87}} = \frac{dE}{do_{87}} \frac{do_{87}}{d\ net_{87}} \frac{d\ net_{87}}{dw_{72,87}}$$

o_{72}

$$\frac{do_{87}}{d\ net_{87}} = \frac{d\phi(net_{87})}{d\ net_{87}} = \phi'(net_{87}) = \phi(net_{87})(1-\phi(net_{87})) = o_{87}(1-o_{87})$$

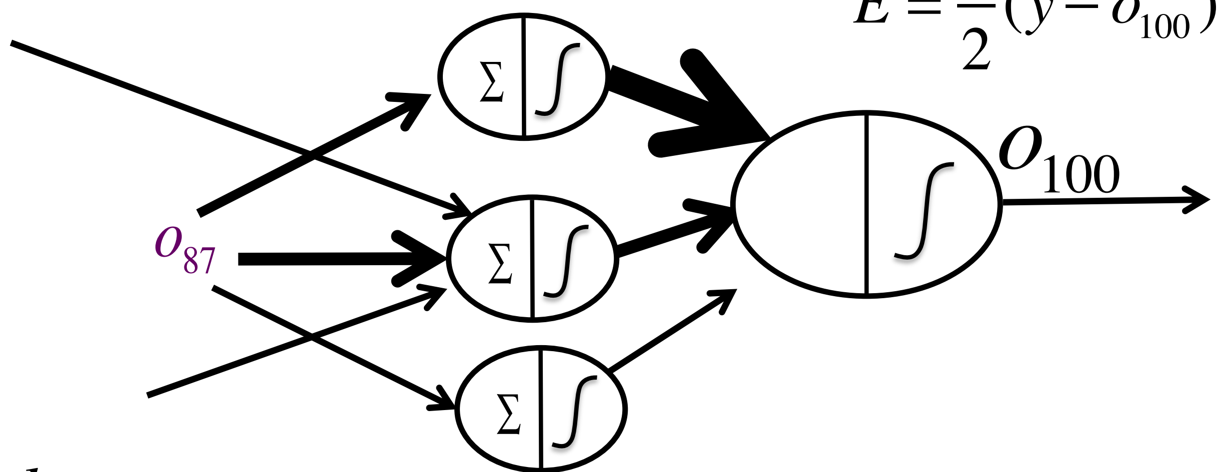


$$\frac{dE}{dw_{72,87}} = \frac{dE}{do_{87}} \frac{do_{87}}{d\ \text{net}_{87}} \frac{d\ \text{net}_{87}}{dw_{72,87}}$$

o_{72}

90's

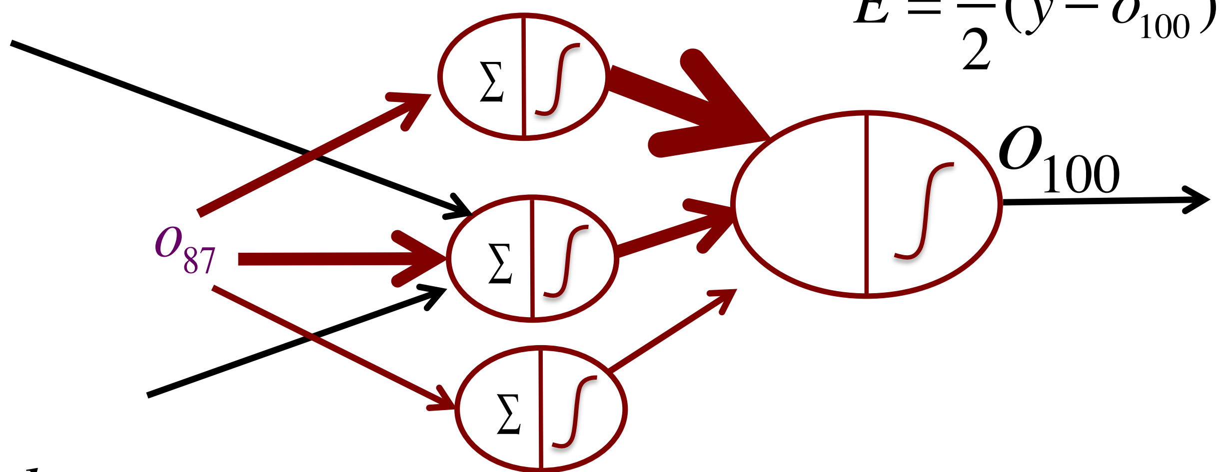
$$E = \frac{1}{2}(y - o_{100})^2$$



$$\frac{dE}{dw_{72,87}} = \frac{dE}{do_{87}} \overbrace{\frac{do_{87}}{d\ \mathrm{net}_{87}}}^{o_{87}(1-o_{87})} \underbrace{\frac{d\ \mathrm{net}_{87}}{dw_{72,87}}}_{o_{72}}$$

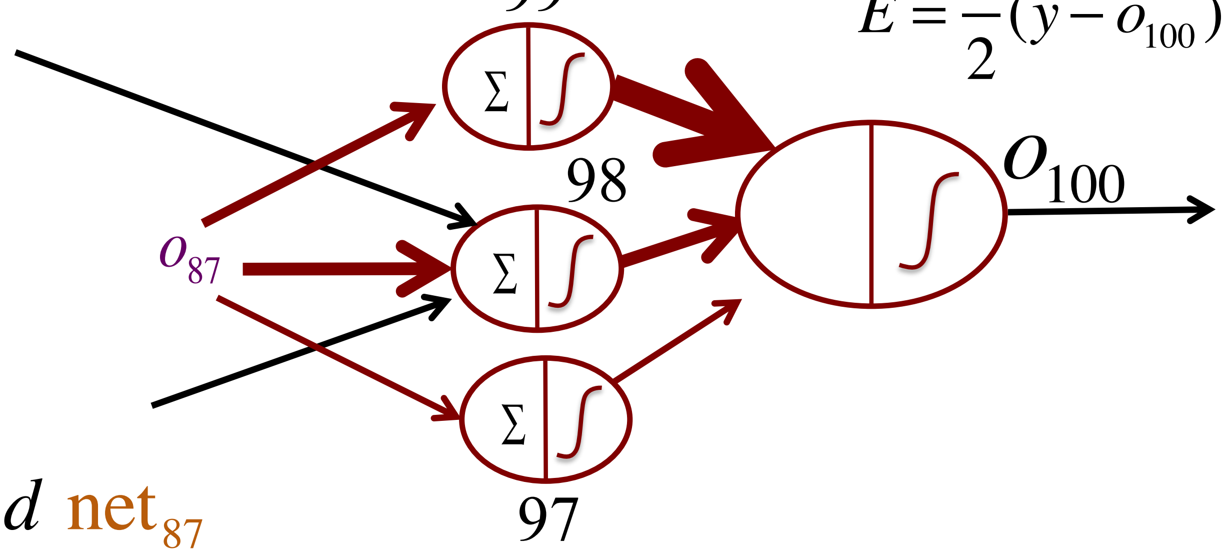
90's

$$E = \frac{1}{2}(y - o_{100})^2$$



$$\frac{dE}{dw_{72,87}} = \frac{dE}{do_{87}} \overbrace{\frac{do_{87}}{d\ \mathrm{net}_{87}}}^{o_{87}(1-o_{87})} \underbrace{\frac{d\ \mathrm{net}_{87}}{dw_{72,87}}}_{o_{72}}$$

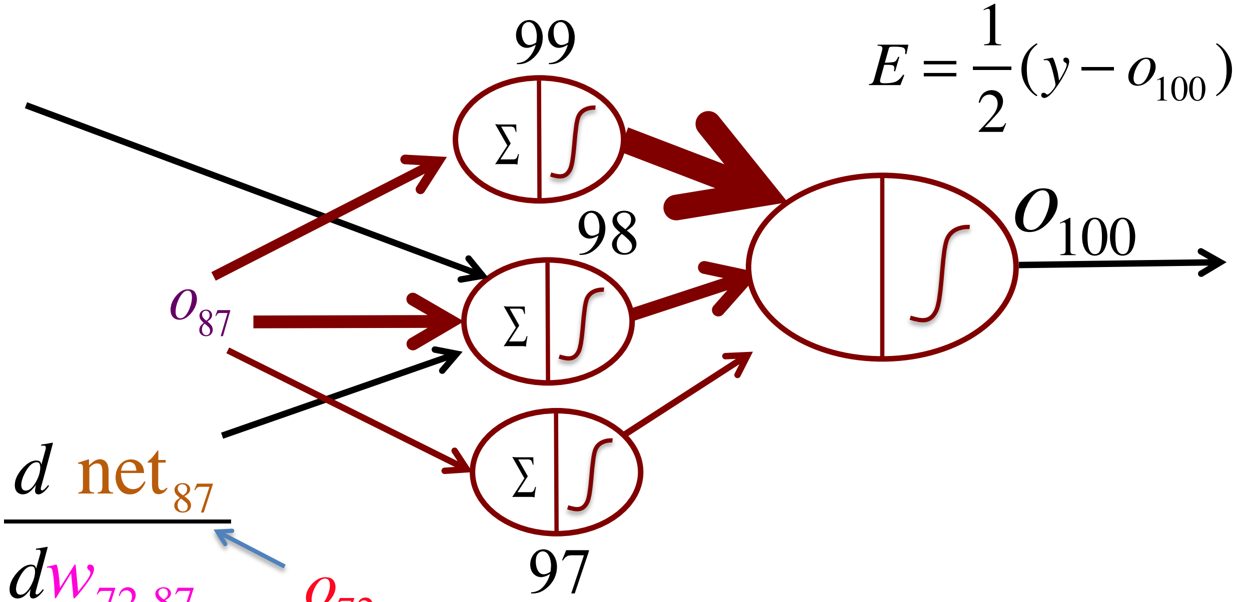
$$E = \frac{1}{2}(y - o_{100})^2$$



$$\frac{dE}{dw_{72,87}} = \frac{dE}{do_{87}} \overbrace{\frac{do_{87}}{d\ \mathrm{net}_{87}}}^{o_{87}(1-o_{87})} \underbrace{\frac{d\ \mathrm{net}_{87}}{dw_{72,87}}}_{o_{72}}$$

$$\frac{dE}{do_{87}} = \frac{dE}{d\ \mathrm{net}_{99}} \frac{d\ \mathrm{net}_{99}}{do_{87}} + \frac{dE}{d\ \mathrm{net}_{98}} \frac{d\ \mathrm{net}_{98}}{do_{87}} + \frac{dE}{d\ \mathrm{net}_{97}} \frac{d\ \mathrm{net}_{97}}{do_{87}} + \ldots$$

$$E = \frac{1}{2}(y - o_{100})^2$$



$o_{87}(1 - o_{87})$

$$\frac{dE}{dw_{72,87}} = \frac{dE}{do_{87}} \frac{do_{87}}{d\ \text{net}_{87}} \frac{d\ \text{net}_{87}}{dw_{72,87}}$$

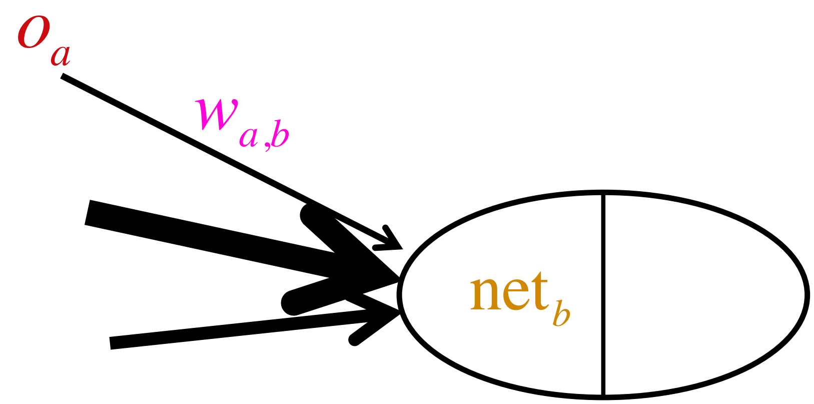
o_{72}

$$\frac{dE}{do_{87}} = \frac{dE}{d\ \text{net}_{99}} \frac{d\ \text{net}_{99}}{do_{87}} + \frac{dE}{d\ \text{net}_{98}} \frac{d\ \text{net}_{98}}{do_{87}} + \frac{dE}{d\ \text{net}_{97}} \frac{d\ \text{net}_{97}}{do_{87}} + \ldots$$

$$= \delta_{99} w_{87,99} + \delta_{98} w_{87,98} + \delta_{97} w_{87,97} + \ldots = \sum_{\ell \in L} \delta_{\ell}\, w_{87,\ell}$$

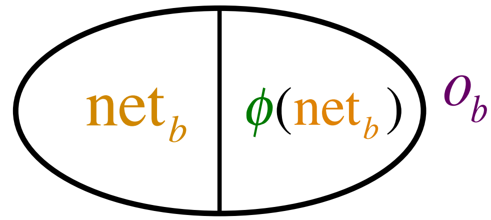
$$\frac{dE}{dw_{a,b}} = \frac{dE}{do_b}\frac{do_b}{d\ \mathrm{net}_b}\frac{d\ \mathrm{net}_b}{dw_{a,b}}$$

$$= \frac{dE}{do_b}\frac{do_b}{d\ \mathrm{net}_b} o_a$$



$$\frac{dE}{dw_{a,b}} = \frac{dE}{do_b}\frac{do_b}{d\ \mathrm{net}_b}\frac{d\ \mathrm{net}_b}{dw_{a,b}}$$

$$= \frac{dE}{do_b}\, o_b(1-o_b)\,o_a$$

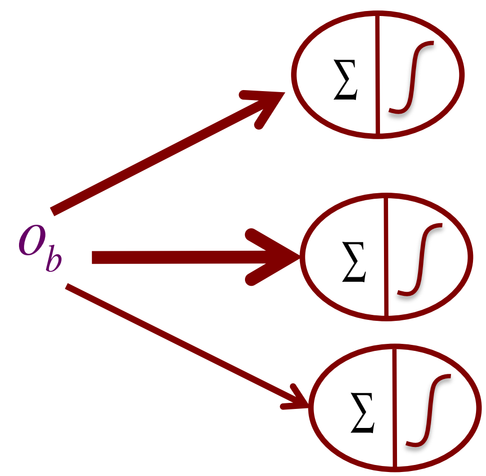


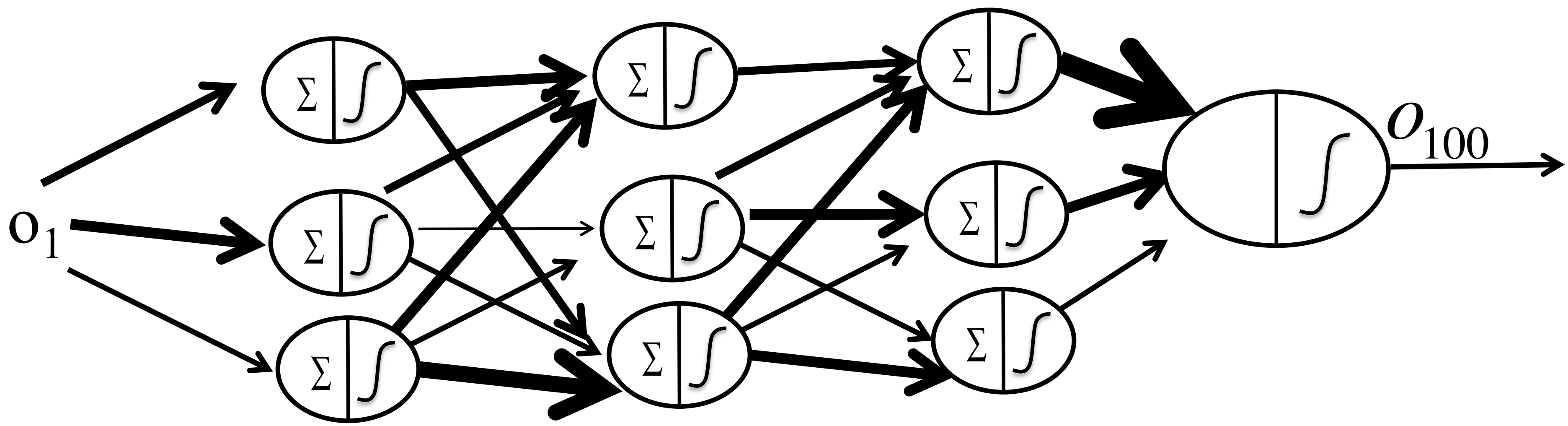
$$\frac{dE}{dw_{a,b}} = \frac{dE}{do_b}\,\frac{do_b}{d\ \mathrm{net}_b}\,\frac{d\ \mathrm{net}_b}{dw_{a,b}}$$

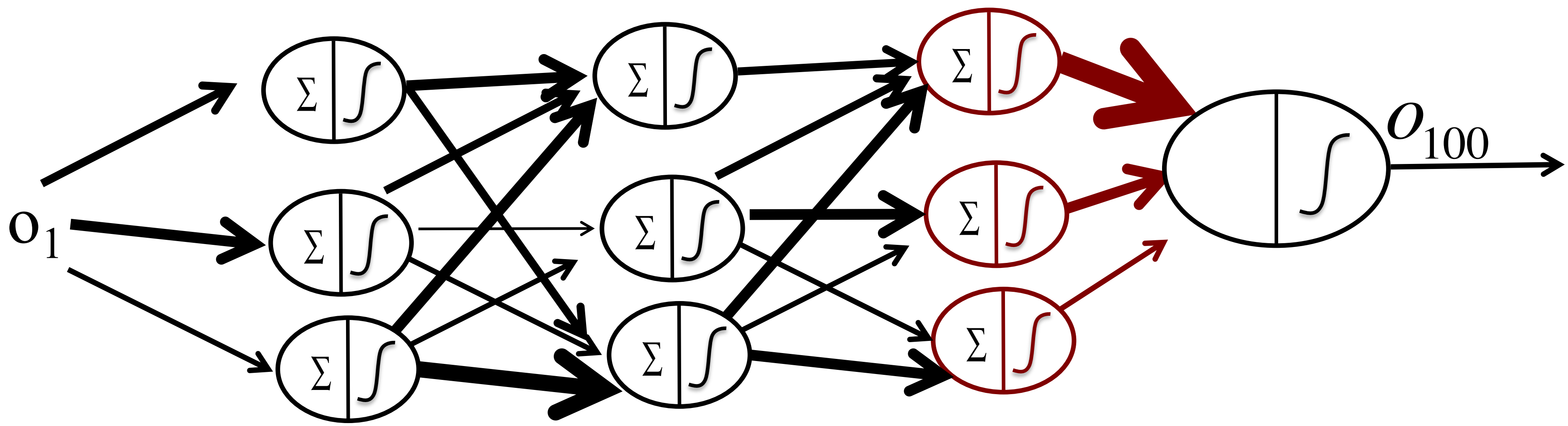
$$= \left(\sum_{\ell \in L} \delta_\ell w_{b,\ell}\right) o_b (1 - o_b) o_a$$

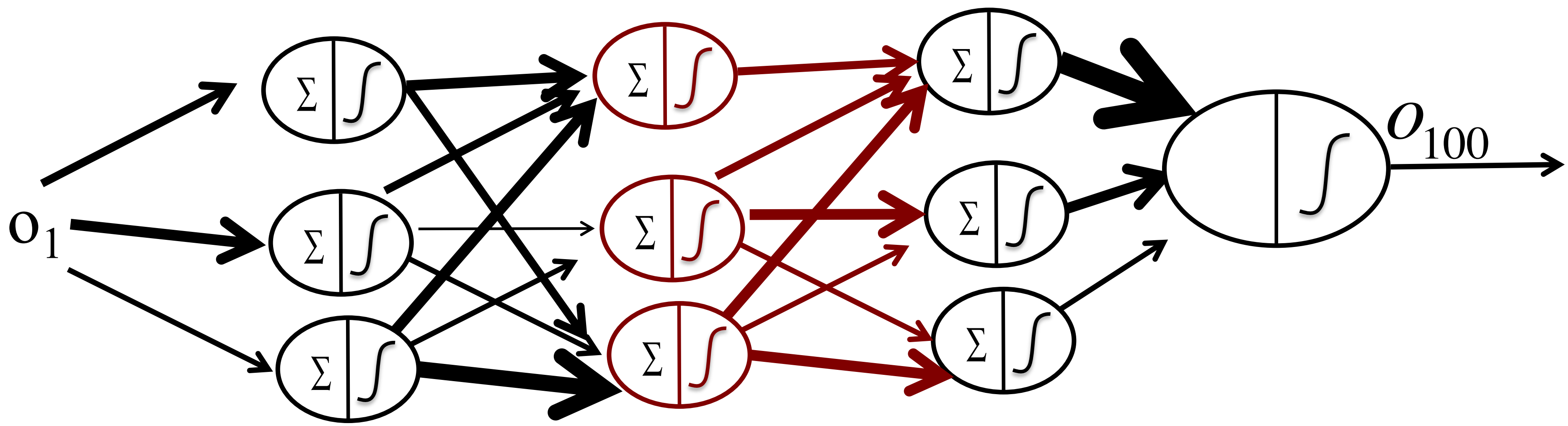
The $\ell's$ are downstream. We must have already computed all the δ_ℓ's ahead of us to compute this.

ℓ's

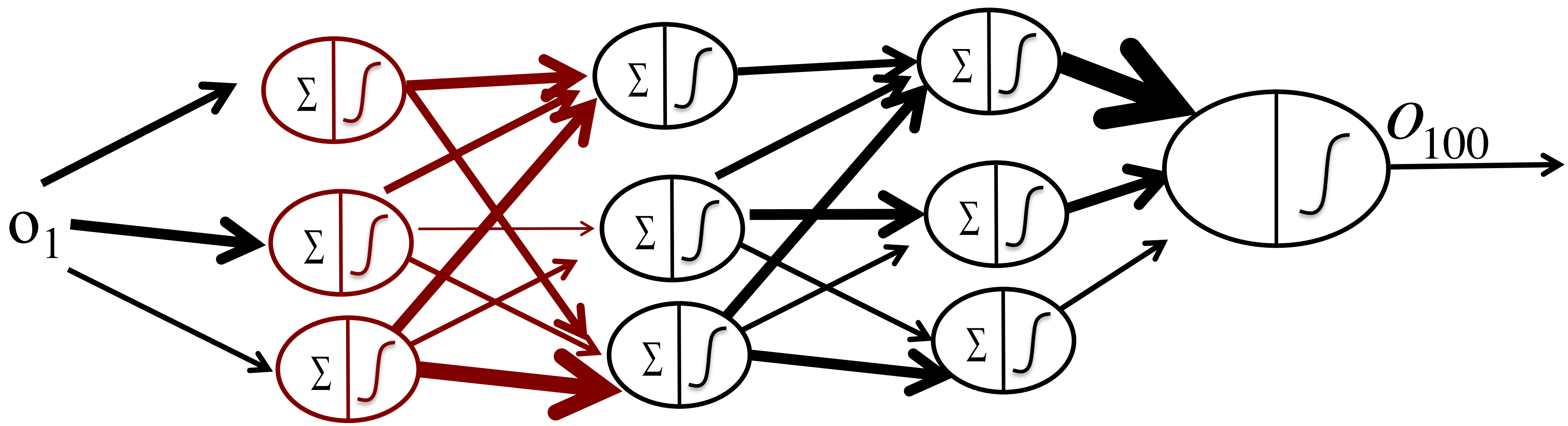



o_1
o_{100}

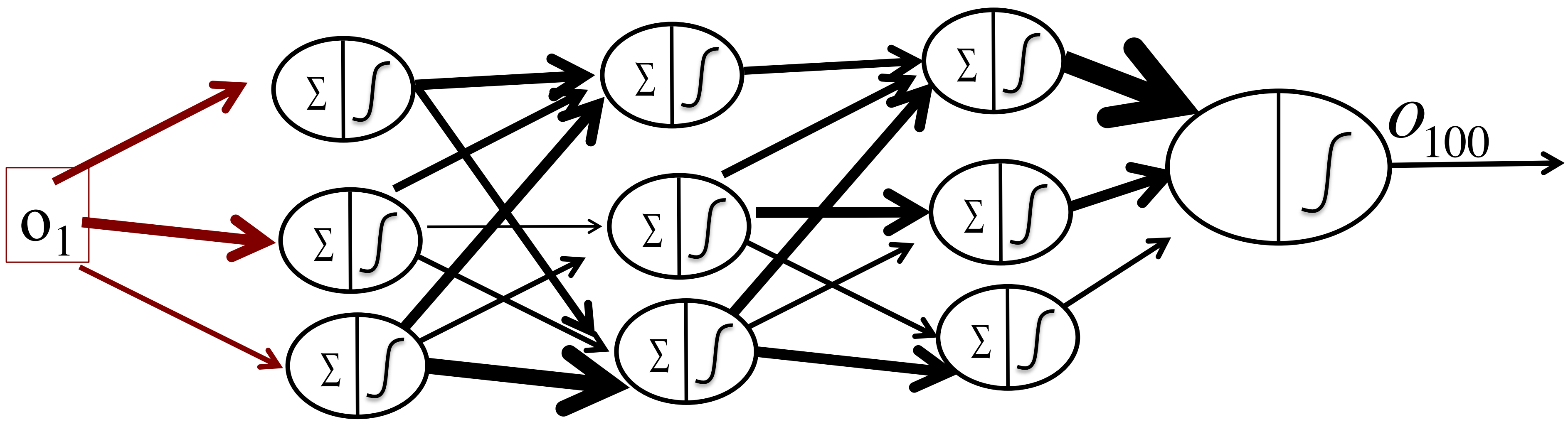

o_1
o_{100}



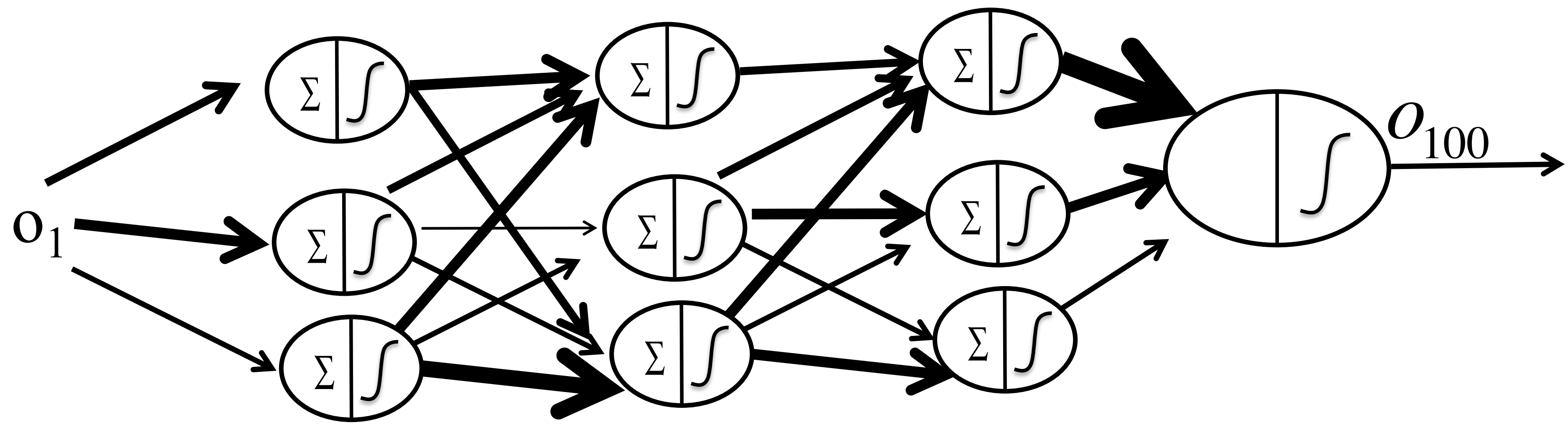
o_1
o_{100}



o1
Σ
o100



o1
Σ
Σ
Σ
Σ
Σ
Σ
Σ
Σ
Σ
o100



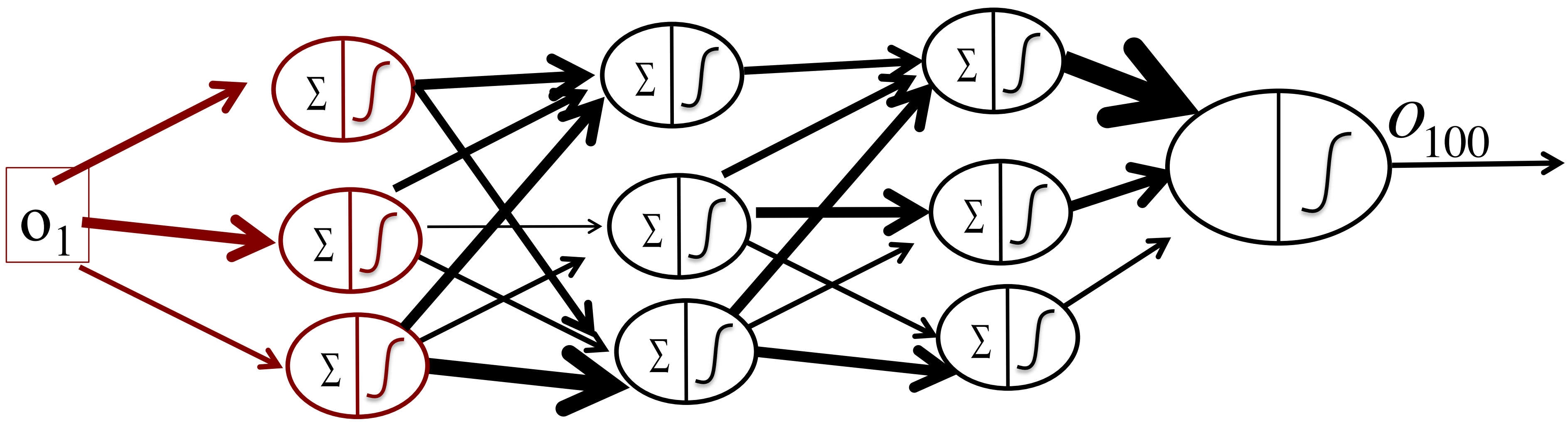
o_1
o_{100}

Backpropagation

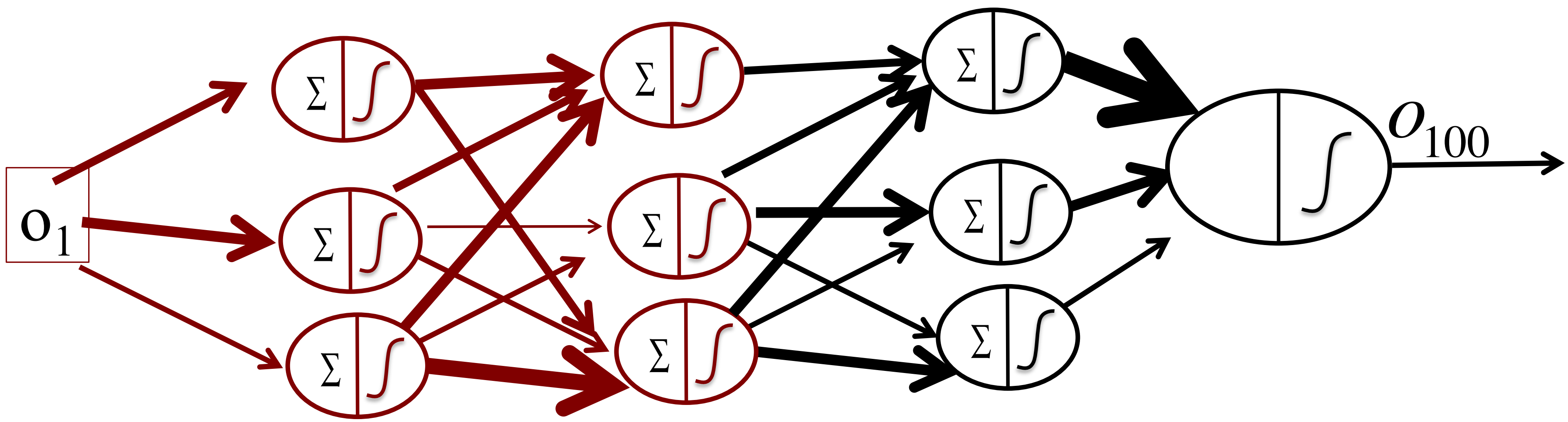
- Now we know how to compute $\frac{dE}{dw_{a,b}}$ for all of the $w_{a,b}$'s.
- Let's do gradient descent.

$$w_{a,b} \longleftarrow w_{a,b} - \alpha \frac{dE}{dw_{a,b}}$$

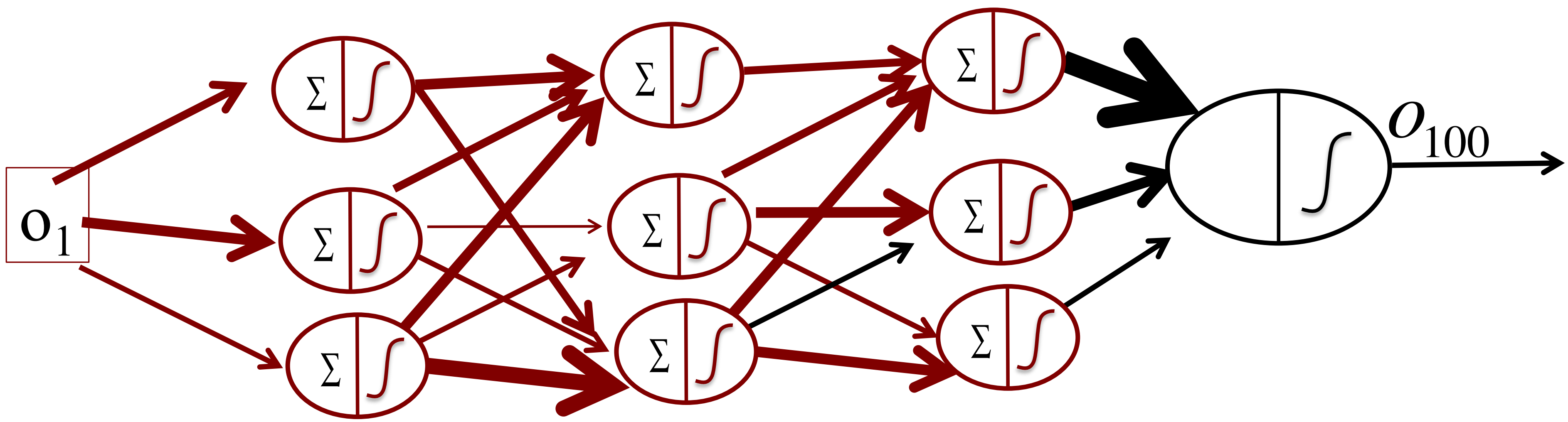
- α is between 0 and 1. Called the "learning rate".
- Now we know how to propagate errors back through the network.
- Remember how to go forward?



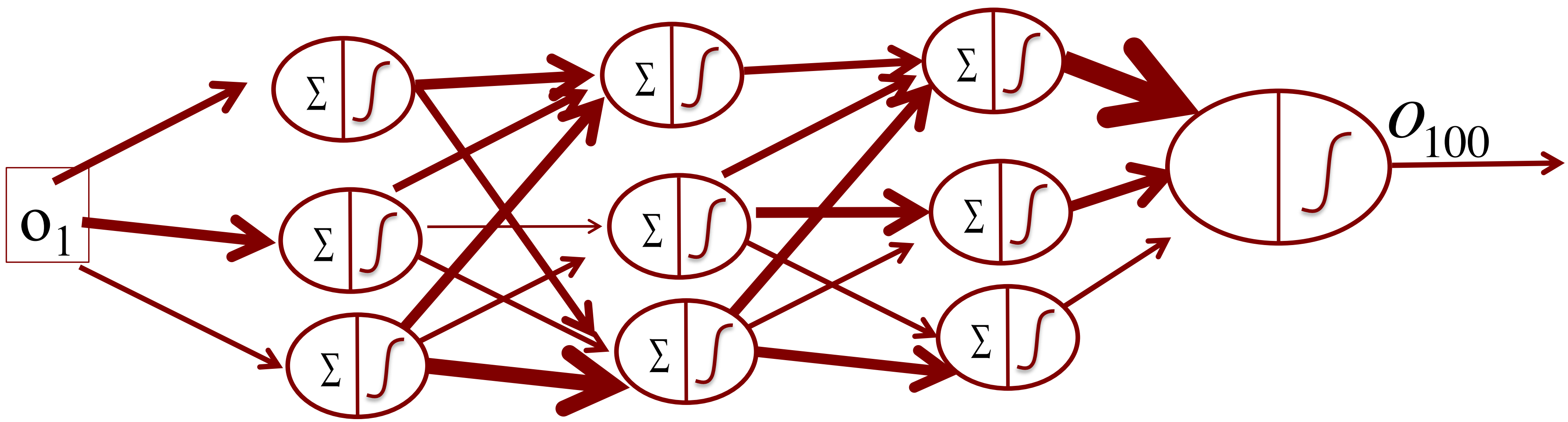
o1
Σ
o100



o_1
o_{100}



o_1
o_{100}

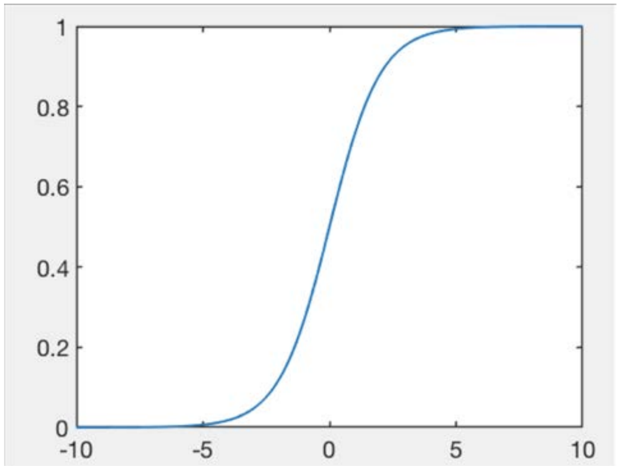

o_1
o_{100}

Backpropagation

- Repeat going backwards (to calculate the gradients), adjusting the weights, and going forwards (to calculate the errors) over and over in order to learn.

Convergence Problems

- NN's have problems with convergence due to vanishing/exploding gradients and saddle points.
- Vanishing gradients come from the flat part of the activation function.
- Exploding gradients happen when we realize that our gradient has vanished and so increase the learning rate and take huge step sizes to compensate (but then mess everything up!)
- Stick to 10^{-5} to 10^{-3} learning rate perhaps?

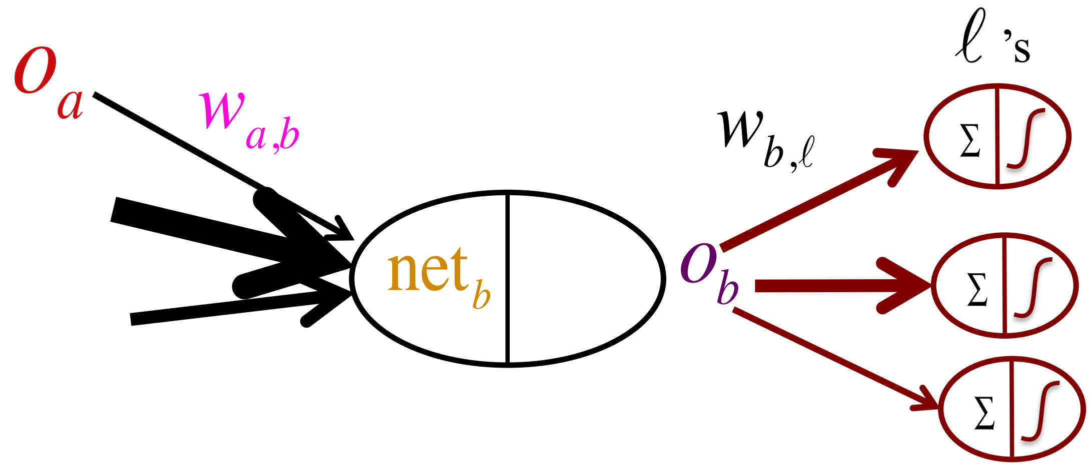


Convergence Problems

- With the sigmoid activation, the derivatives of the input weights for each node are always either all positive or all negative. This is a limitation.

$$\frac{dE}{dw_{a,b}} = \frac{dE}{do_b}\frac{do_b}{d\ \text{net}_b}\frac{d\ \text{net}_b}{dw_{a,b}}$$

$$= \underbrace{\left(\sum_{\ell \in L} \delta_\ell w_{b,\ell}\right)}_{\text{does not depend on a}} \underbrace{o_b(1-o_b)o_a}_{\text{positive, since all outputs of sigmoid are between 0 and 1.}}$$



Convergence Problems

- Bottom line – most people do not use sigmoid-like activation functions, even though this is more biologically relevant.

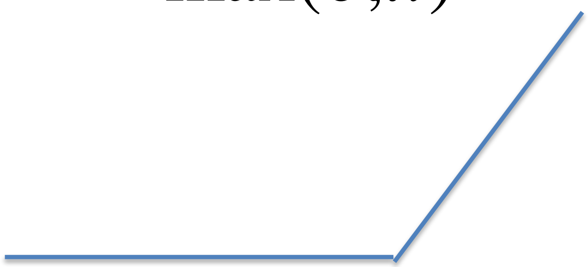
Sigmoid $$\sigma(x) = \frac{1}{1+e^{-x}}$$

Hyperbolic tangent $$\tanh(x)$$

Convergence Problems

Rectified Linear Unit (ReLU)

$$\max(0,x)$$



Removes vanishing gradients when nodes are "activated," meaning x>0.

(Krizhevsky et al., 2012)

Leaky ReLU

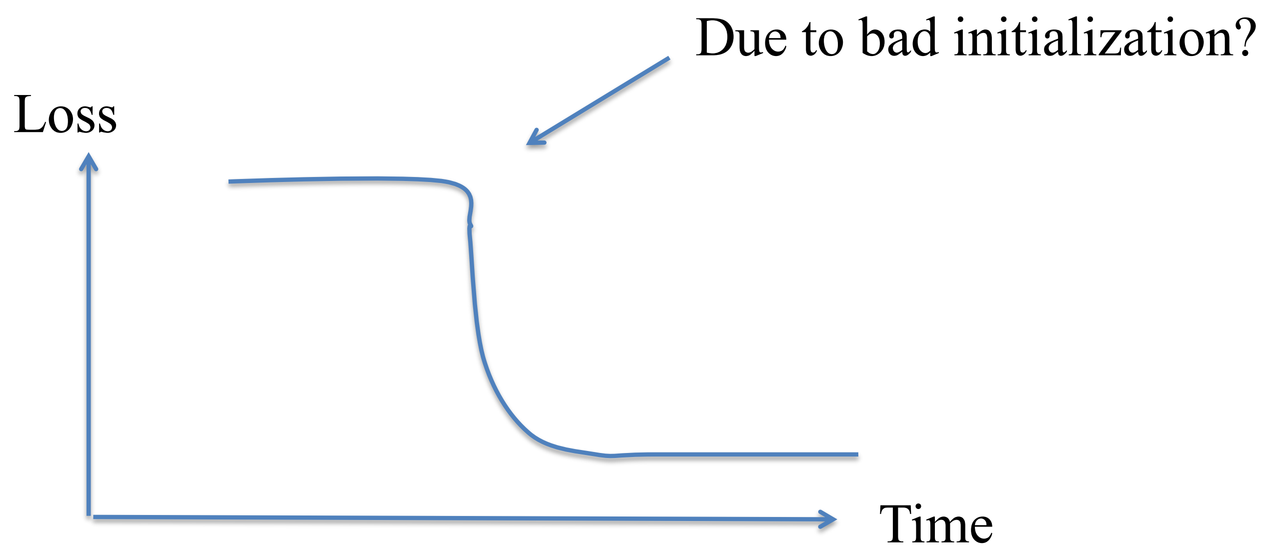
$$\max(0.1x,x)$$

Removes vanishing gradients, but prefers that non-activated nodes be as "non-activated" as possible, which doesn't make much sense.

(Mass et al., 2013; He et al., 2015)

Convergence Problems

- Initialization of the networks weights is really important. I have no idea how to do it.



Convergence Problems

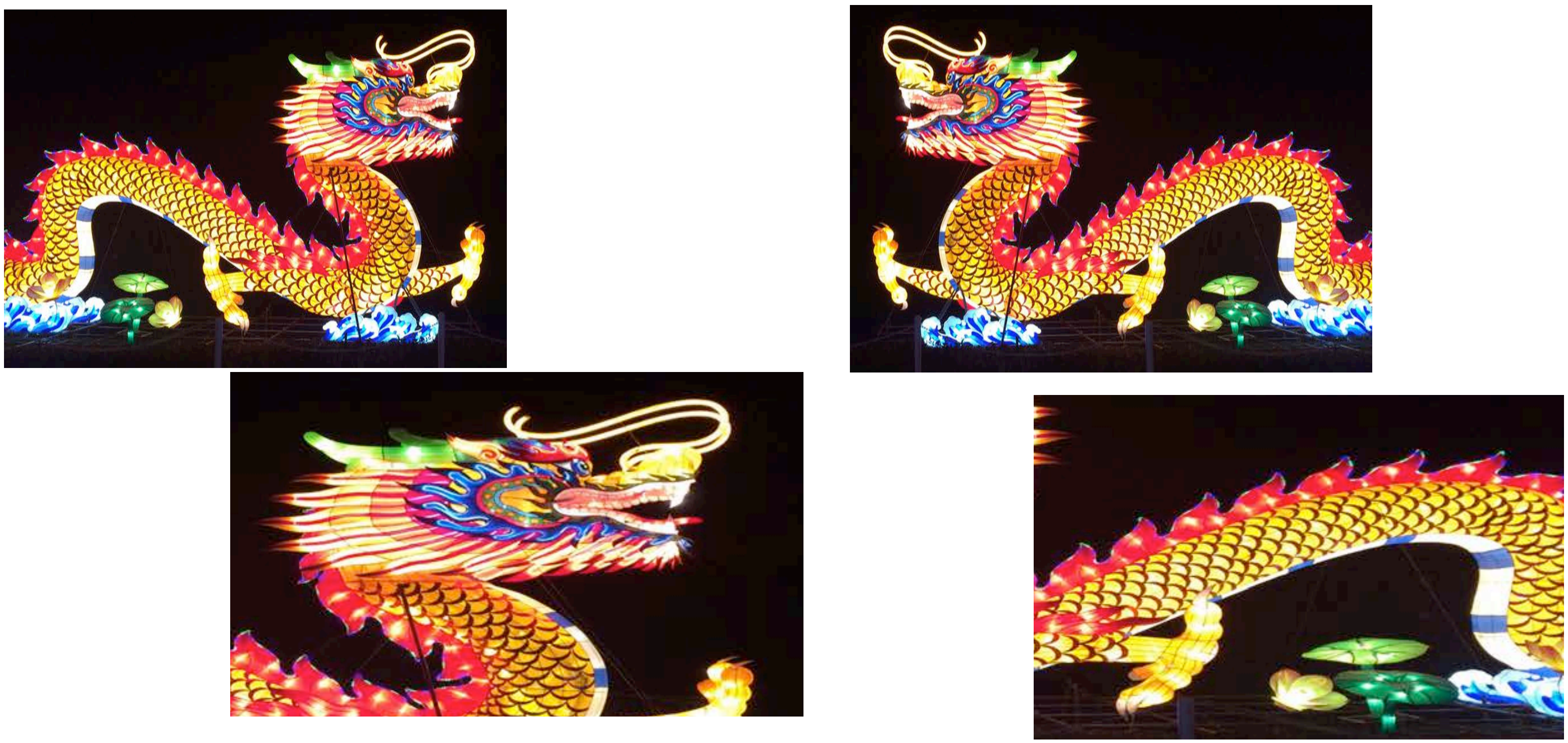
- Batch Normalization (Ioffe and Szegedy, 2015) is a step in a NN that:
  - Normalizes the outputs o_i of several nodes (a "mini-batch") in the same layer (as usual, subtract the mean of the o_i's divide by their standard deviation).
  - Includes the mean and standard deviation as separate parameters to be learned.
  - Usually the normalization is before the nonlinear activation function.
  - This adds regularization and helps gradient flow in the network but sometimes it messes things up.

Data Augmentation

Chinese Lantern Festival, Cary NC, 2017

Data Augmentation

- include artificial data, such as horizontal flips, rotations, resized, cropped training images, change contrast and brightness, distortion, etc.



Chinese Lantern Festival, Cary NC, 2017

Convolutional NN's

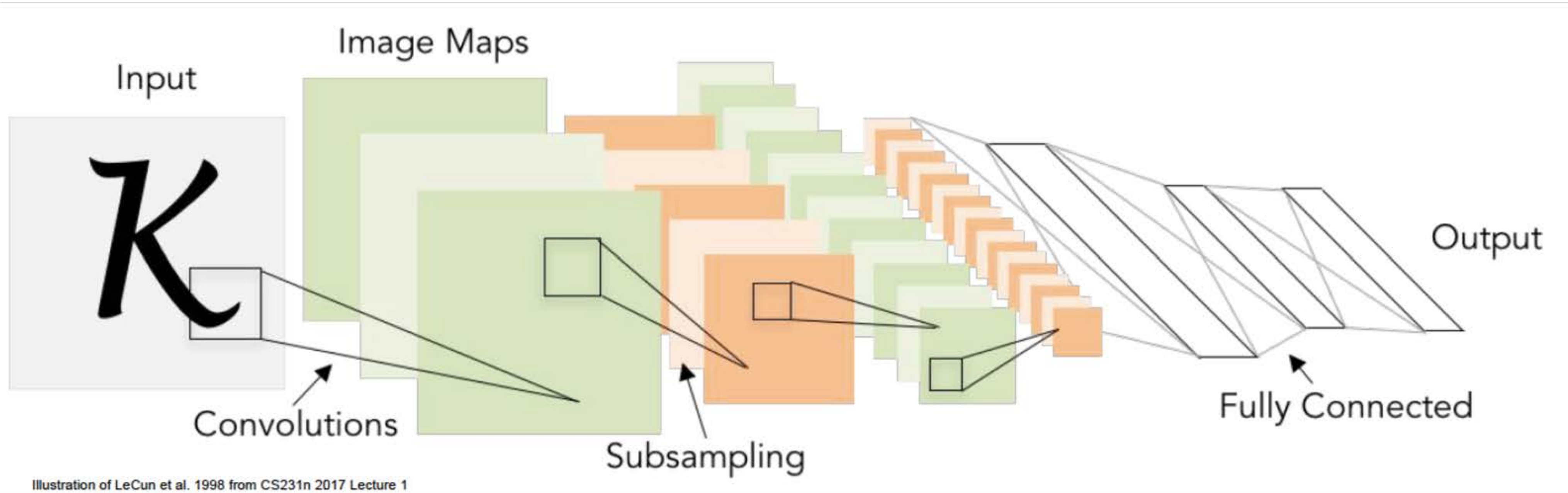
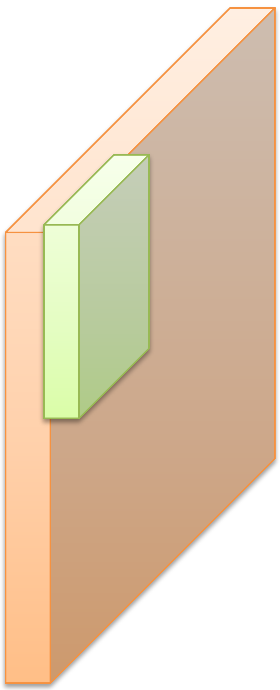


Image from LeCun et al 1998, reproduced also from Li, Johnson, Yeung, 2017

Convolutional NN's

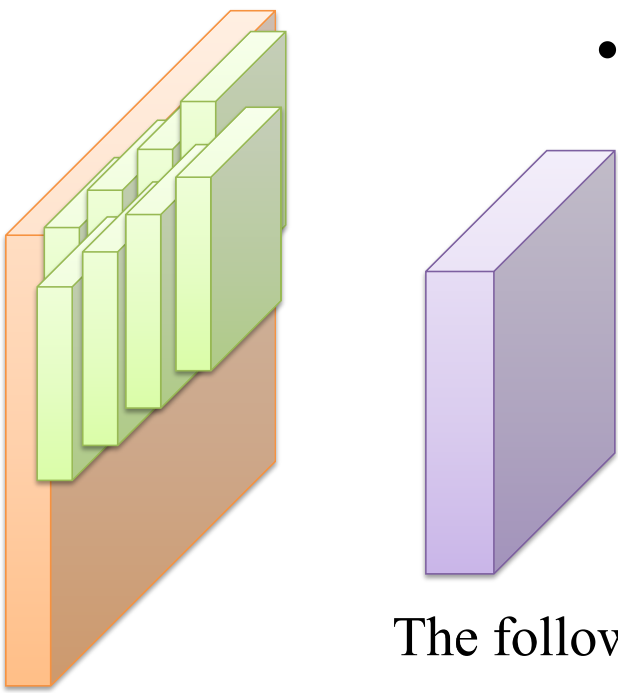
- Convolve means to slide the filter over all spatial locations, and sum up the filter weights times the input.



- An edge filter will detect edges.

Convolutional NN's

- Convolve means to slide the filter over all spatial locations, and sum up the filter weights times the input.

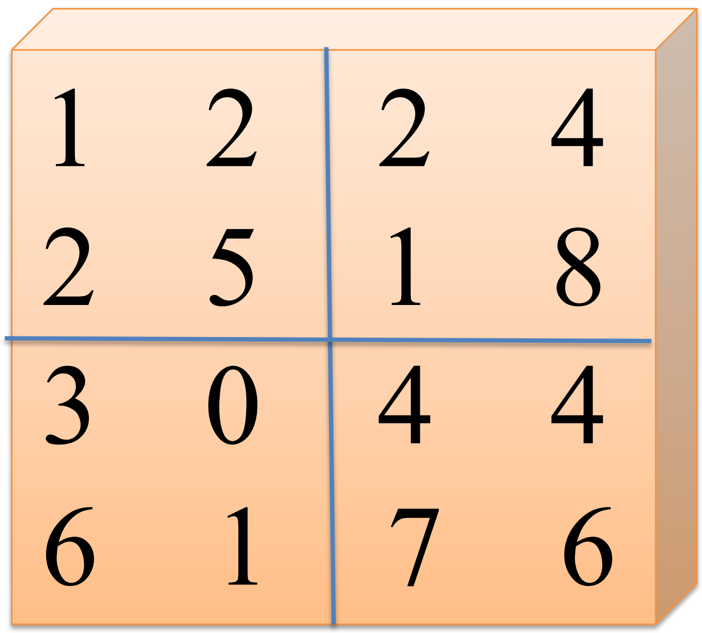


- Stride of 5 means we step by 5's when we convolve.

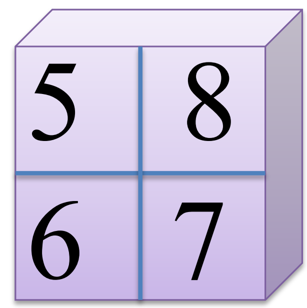
The following layer is smaller by a factor of 5.

Convolutional NN's

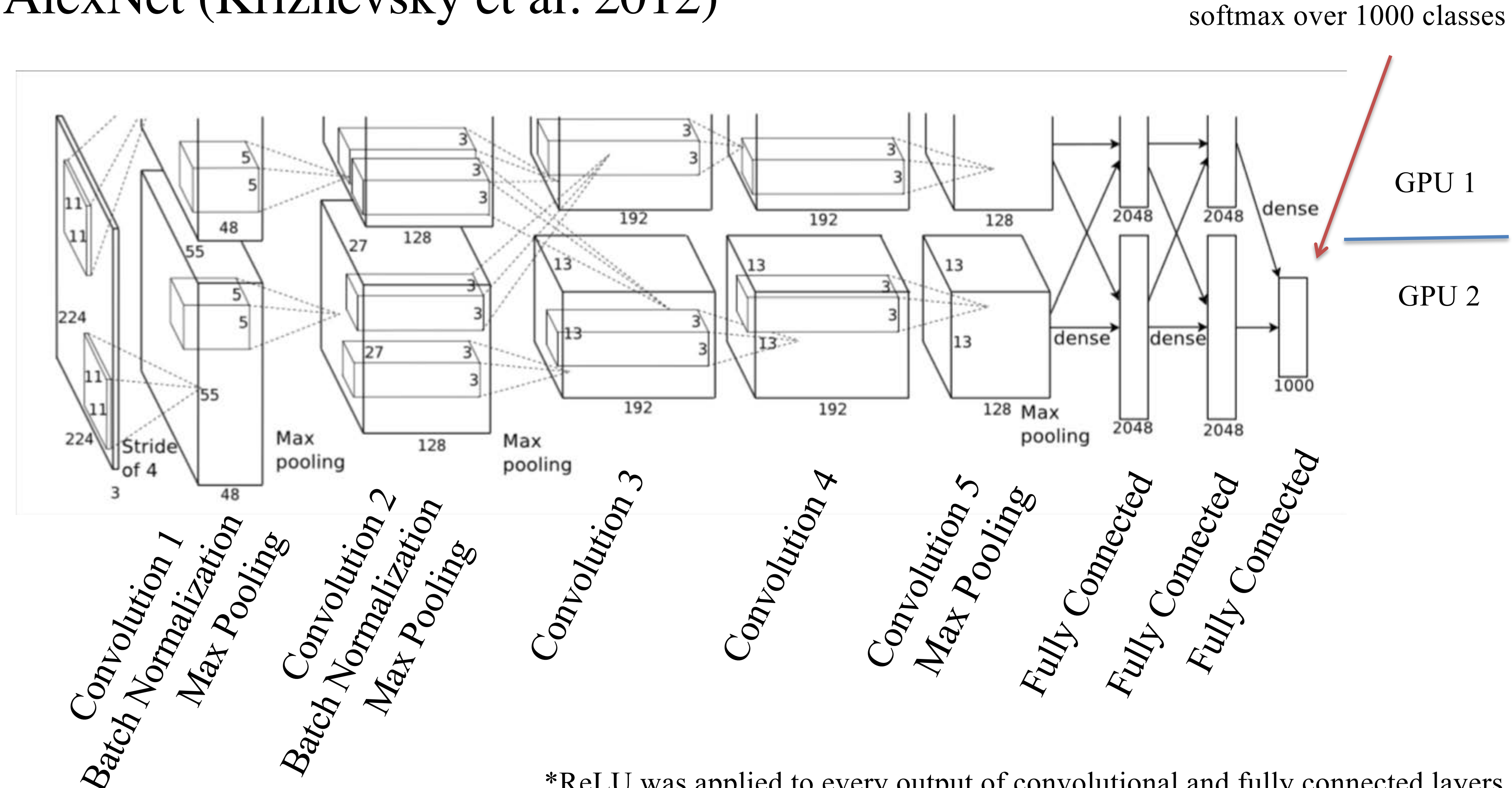
- Max pooling means to convolve with a max function.
- Intuitively keeps track of whether an earlier filter has detected something.



2 x 2 max pool filter and stride 2

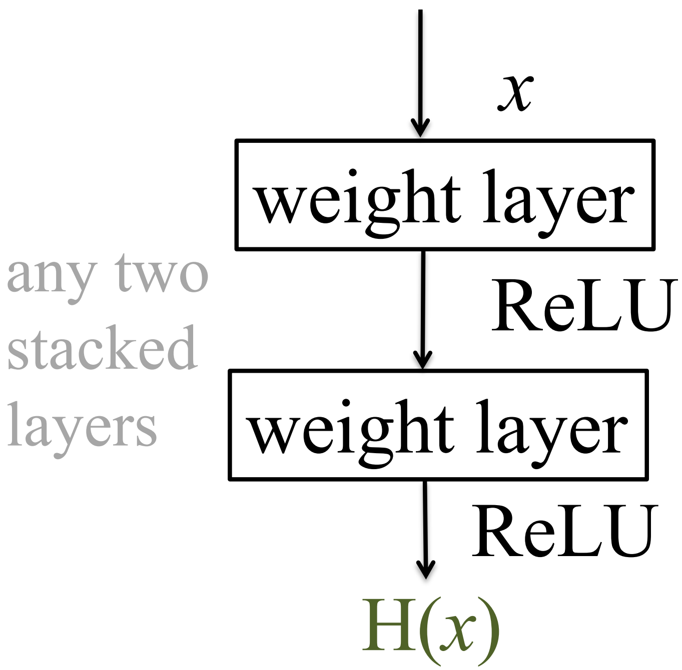


• AlexNet (Krizhevsky et al. 2012)



*ReLU was applied to every output of convolutional and fully connected layers

Residual Nets (He et al., 2016)

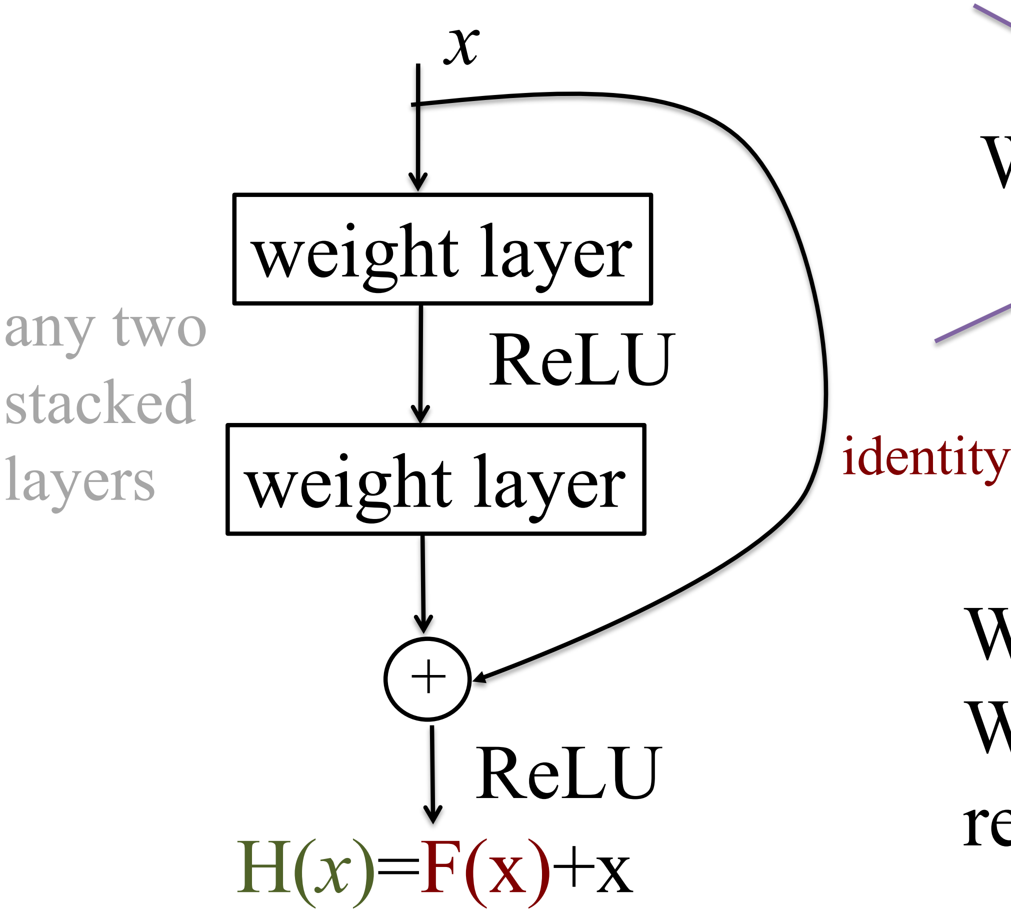


~~We hope to fit H(x).~~

Slides recreated from Kaiming He's tutorial

http://kaiminghe.com/icml16tutorial/icml2016_tutorial_deep_residual_networks_kaiminghe.pdf

Residual Nets



~~We hope to fit $H(x)$.~~

We hope to fit $F(x)$.
We are now learning a residual of identity.

http://kaiminghe.com/icml16tutorial/icml2016_tutorial_deep_residual_networks_kaiminghe.pdf

Residual Nets

- By adding x, the derivative of the error with respect to x increases by 1. Thus, less vanishing derivatives.

- Allowed networks to go much deeper than before. "From 10 to 1000 layers"

$$H(x)=F(x)+x$$

http://kaiminghe.com/icml16tutorial/icml2016_tutorial_deep_residual_networks_kaiminghe.pdf

Dropout (Srivastava et al., JMLR 2014)

- In each forward pass, for each neuron, with probability p, set all of its output weights to 0.
- p is a hyperparameter, usually p = 0.5.
- During testing, use all nodes.

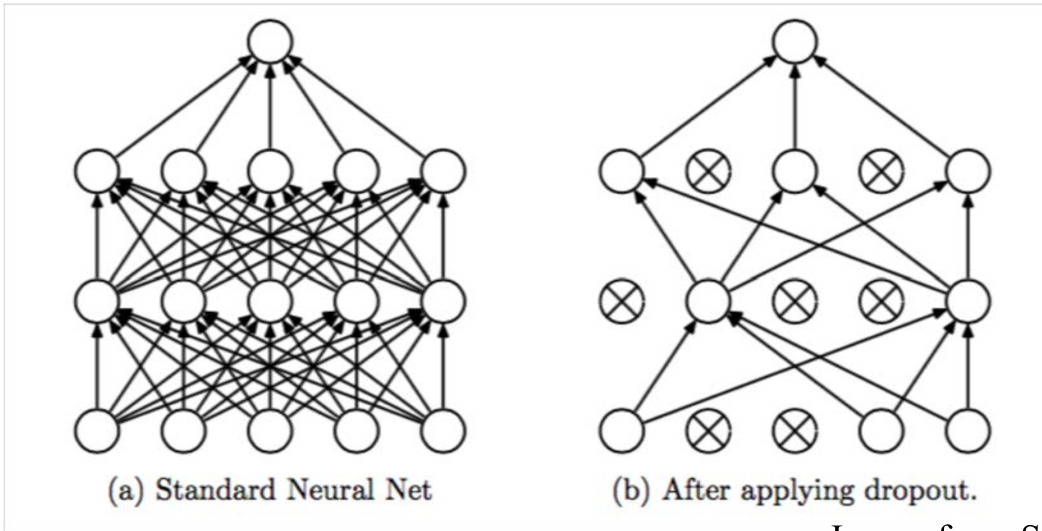


Image from Srivastava et al JMLR 2014)

Dropout (Srivastava et al., JMLR 2014)

- As if we are training exponentially many “sub” models. Similar idea to bagging (averaging many separately trained models together).
- Creates a redundant encoding.

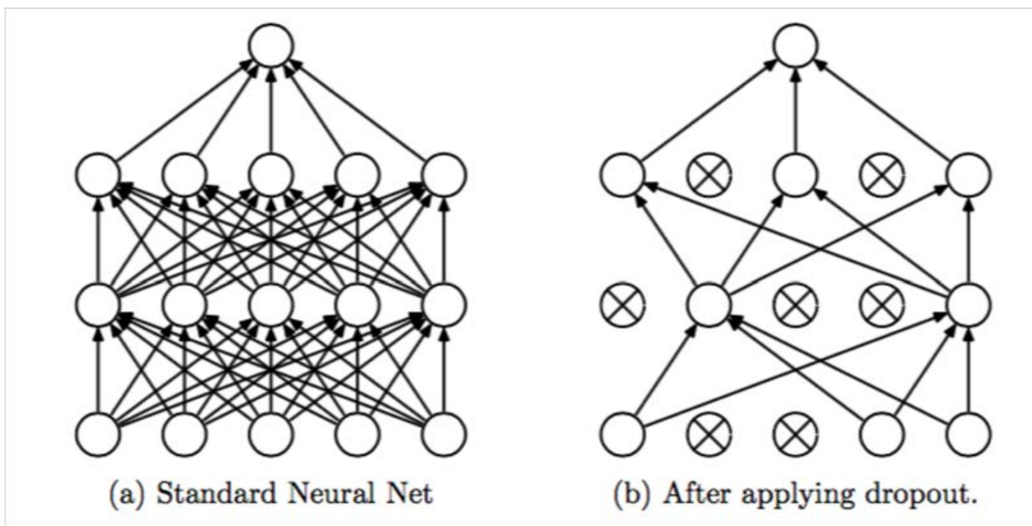
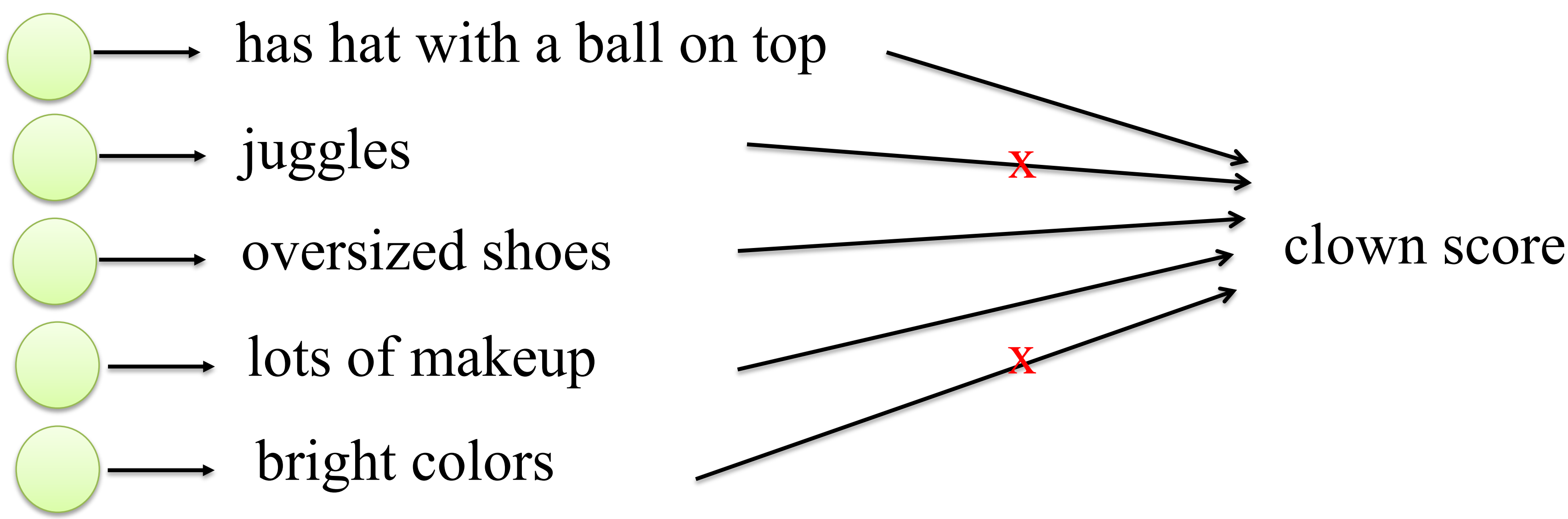


Image from Srivastava et al JMLR 2014)

Dropout (Srivastava et al., JMLR 2014)

- As if we are training exponentially many "sub" models. Similar idea to bagging (averaging many separately trained models together).
- Creates a redundant encoding.



A Big Bag of Tricks

- Dropout
- Batch Normalization
- Data Augmentation
- Residual Networks
- Convolutional Networks
- Activation Functions (ReLU, Leaky ReLU)
- Initialization
- Max Pooling
- :

RNN’s have recently been surpassed by Transformers

- RNN’s used to be the standard for machine translation (sequence-to-sequence). These were useful because they kept a memory of longer-term relationships between words.
- The new standard is the transformer, which places several “multi-head attention nodes” together. The transformer compares all words to all other words in the sentence. It is not recursive.

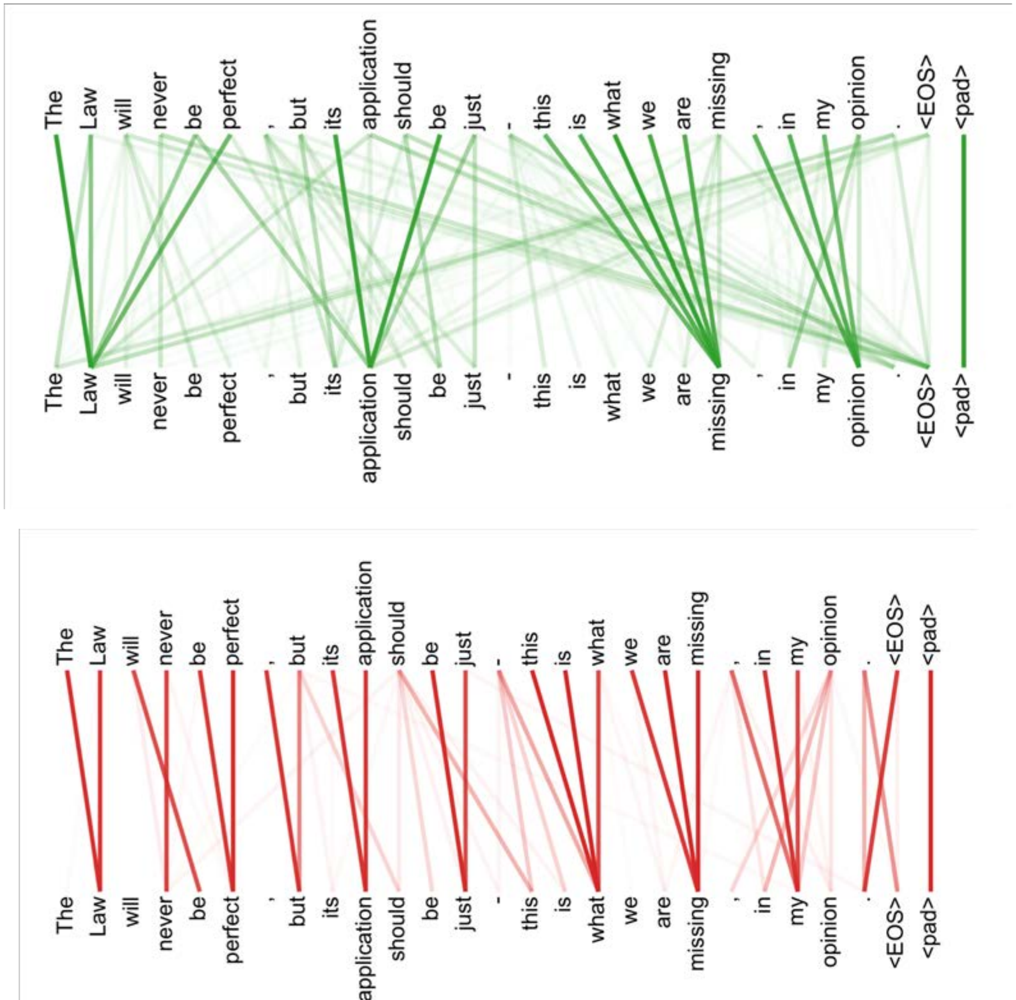


Figure 5: Many of the attention heads exhibit behaviour that seems related to the structure of the sentence. We give two such examples above, from two different heads from the encoder self-attention at layer 5 of 6. The heads clearly learned to perform different tasks.

“Attention is all you need” Vaswani et al 2017

GANS – Generative Adversarial Networks

From Goodfellow et al 2014:

In other words, D and G play the following two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \qquad (1)$$

- GANS are actor-critic models
- They produce realistic-looking images/data (some ethical/legal concerns)
- Used commonly for AI artwork

Is artificial intelligence set to become art's next medium?

AI artwork sells for $432,500 — nearly 45 times its high estimate — as Christie's becomes the first auction house to offer a work of art created by an algorithm

Figure adapted from L. Gatys et al. "A Neural Algorithm of Artistic Style" (2015) by Google AI Blog

Neural networks

- Advantages:
  - highly expressive nonlinear models
  - have advances in computer vision and speech that other methods have not achieved
  - can capture latent structure within the hidden layers
- Disadvantages
  - can get stuck in local optima, could produce bad solutions
  - black box
  - lots of tuning parameters (e.g., the structure of the network)
- There is a recent bag of tricks

Thanks