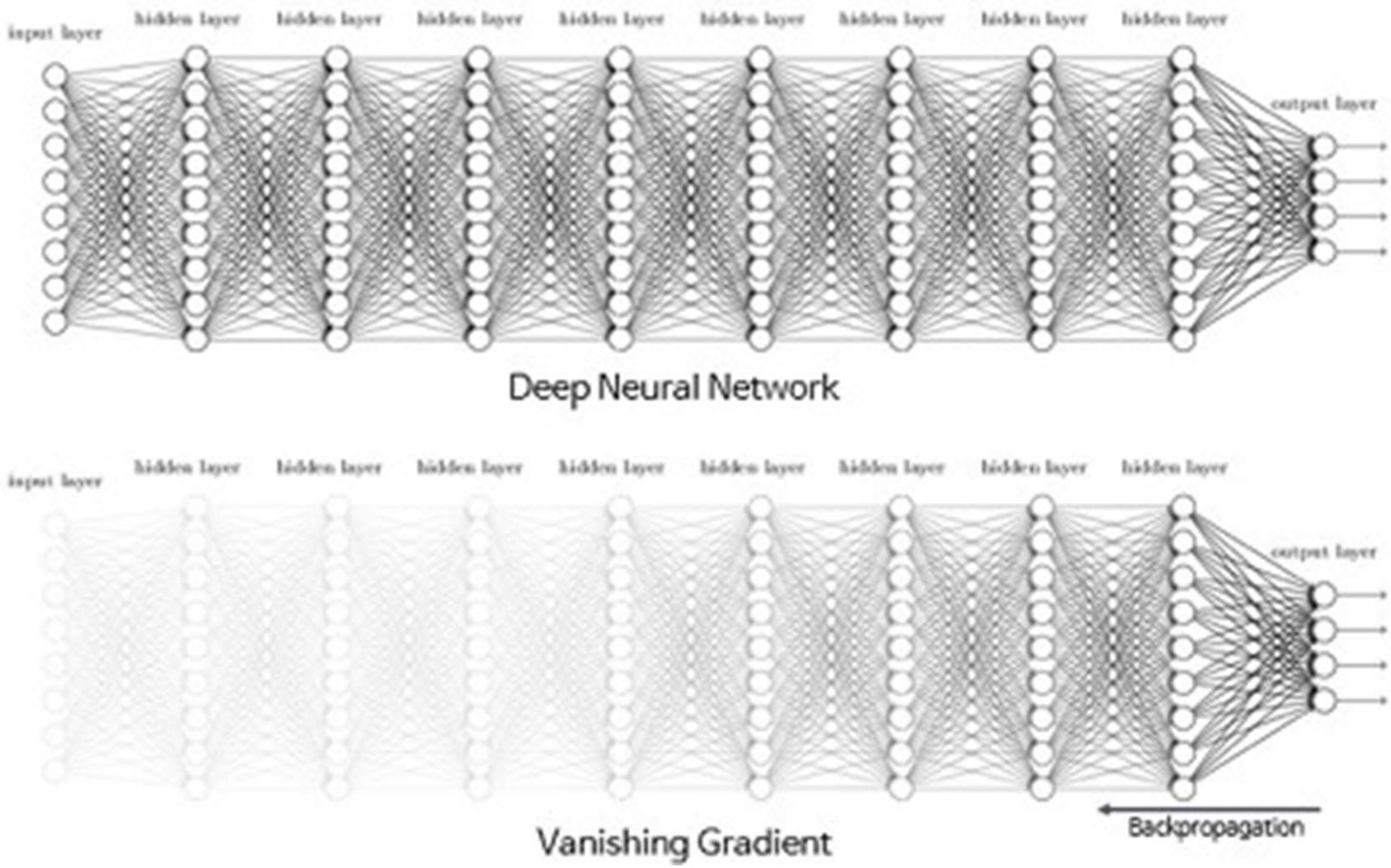
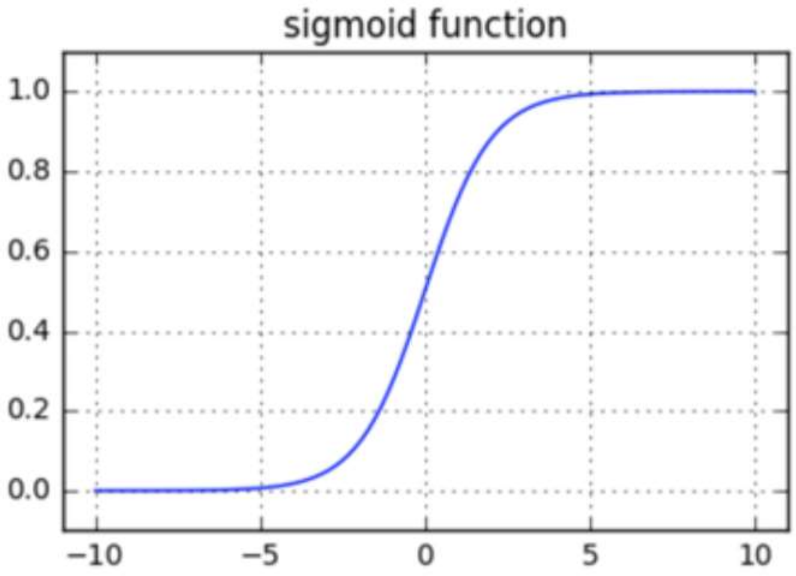
Chapter 5 Deep Learning

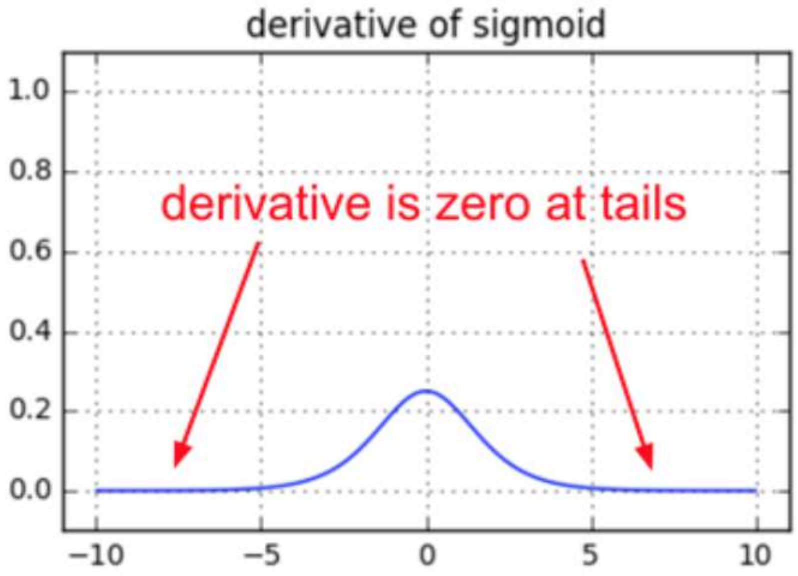
Problem: Vanishing Gradient



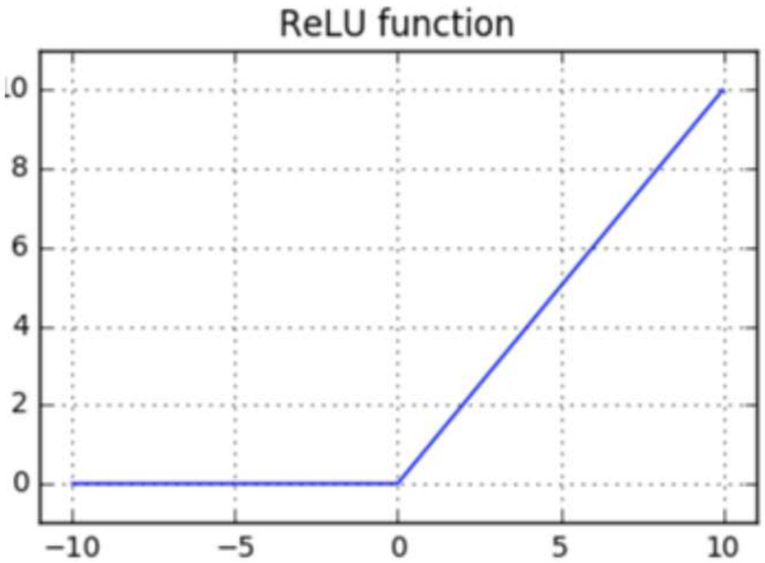
Why Vanishing Gradients

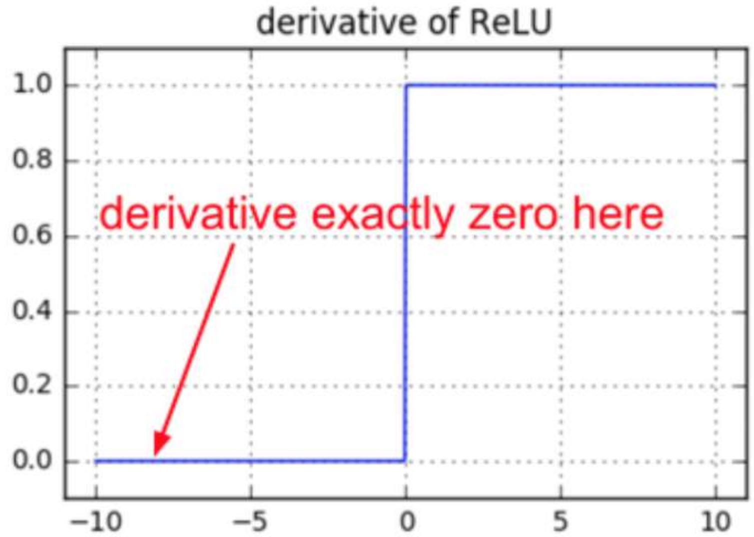
- Gradients of the loss function approach to zero, making the network hard to train.
- Sigmoid function squishes a large input space into a small input space between 0 and 1.
- Therefore, a large change in the input of the sigmoid function will cause a small change in the output.
- Hence, the derivative becomes small.





ReLU and Derivatives

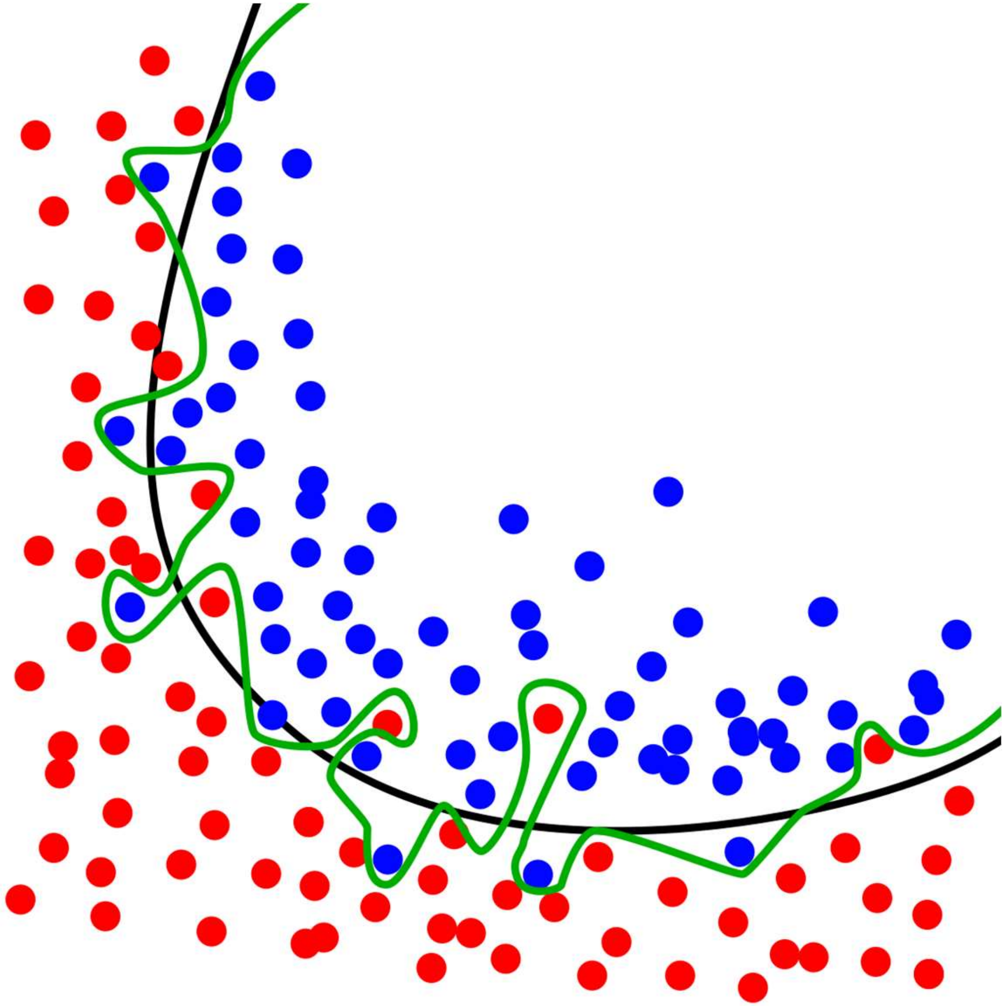




How Does ReLU Solve Vanishing Gradient?

- **RELU** activation **solves** this by having a **gradient** slope of 1, so during backpropagation, there isn't **gradients** passed back that **are** progressively getting smaller and smaller, but instead they **are** staying the same, which **is** how **RELU** **solves** the **vanishing gradient** problem

Overfitting

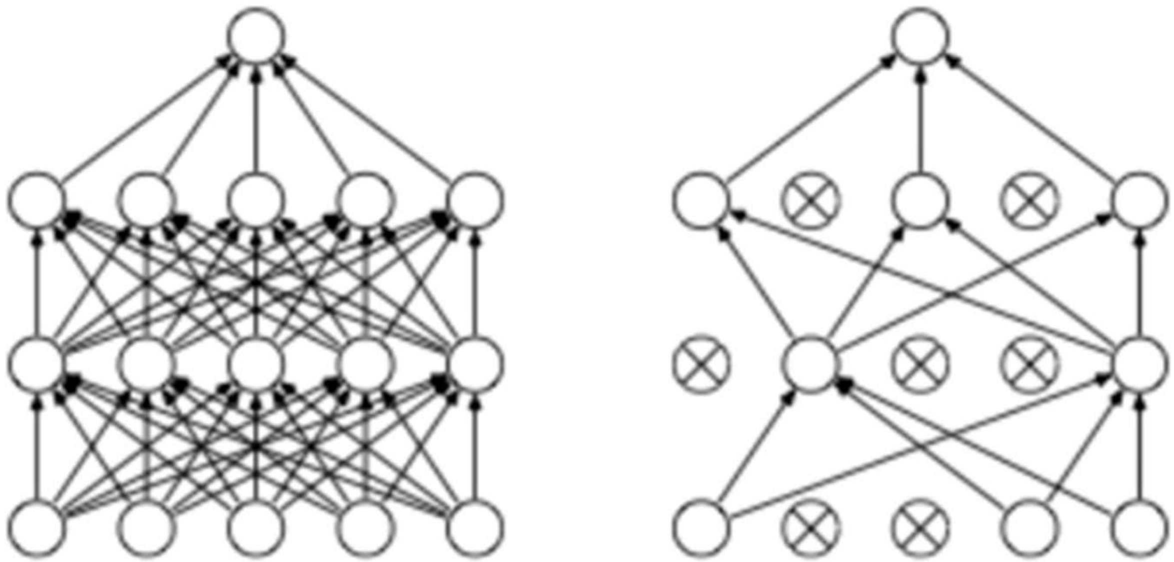


How Does Dropout Reduce Overfitting?

- **Dropout** prevents **overfitting** due to a layer's "over-reliance" on a few of its inputs. Because these inputs aren't always present during training (i.e. they **are** dropped at random), the layer learns to use all of its inputs, improving generalization

Dropout

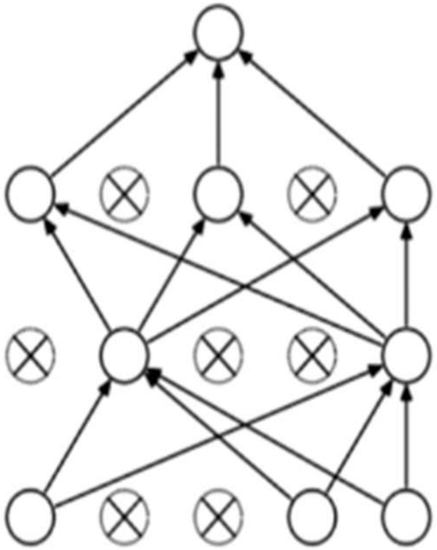
Dropout: A Simple Way to Prevent Neural Networks from Overfitting [Srivastava et al. 2014]



(a) Standard Neural Net (b) After applying dropout.

Dropout

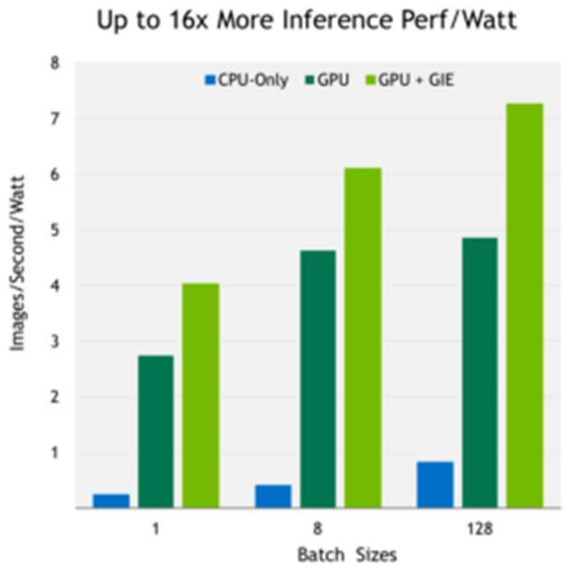
Waaaait a second…
How could this possibly be a good idea?



Forces the network to have a redundant representation.

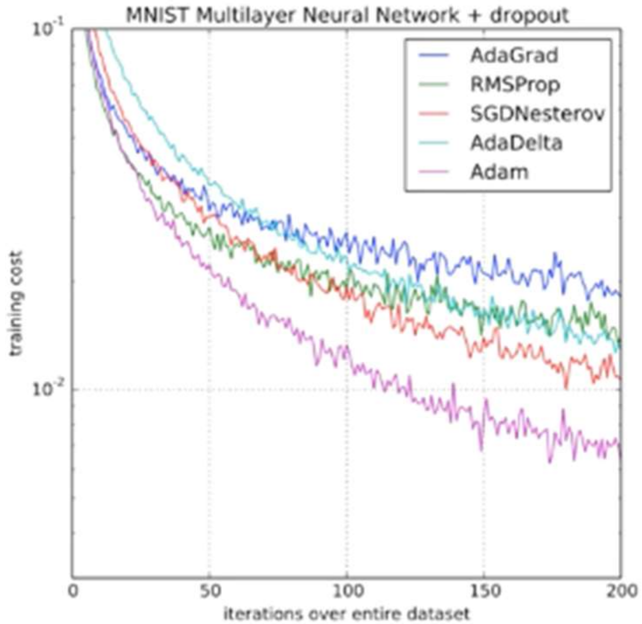
Computational Load

- Multidimensional Optimization
- GPU Computation (GIE, GPU Inference Engine)



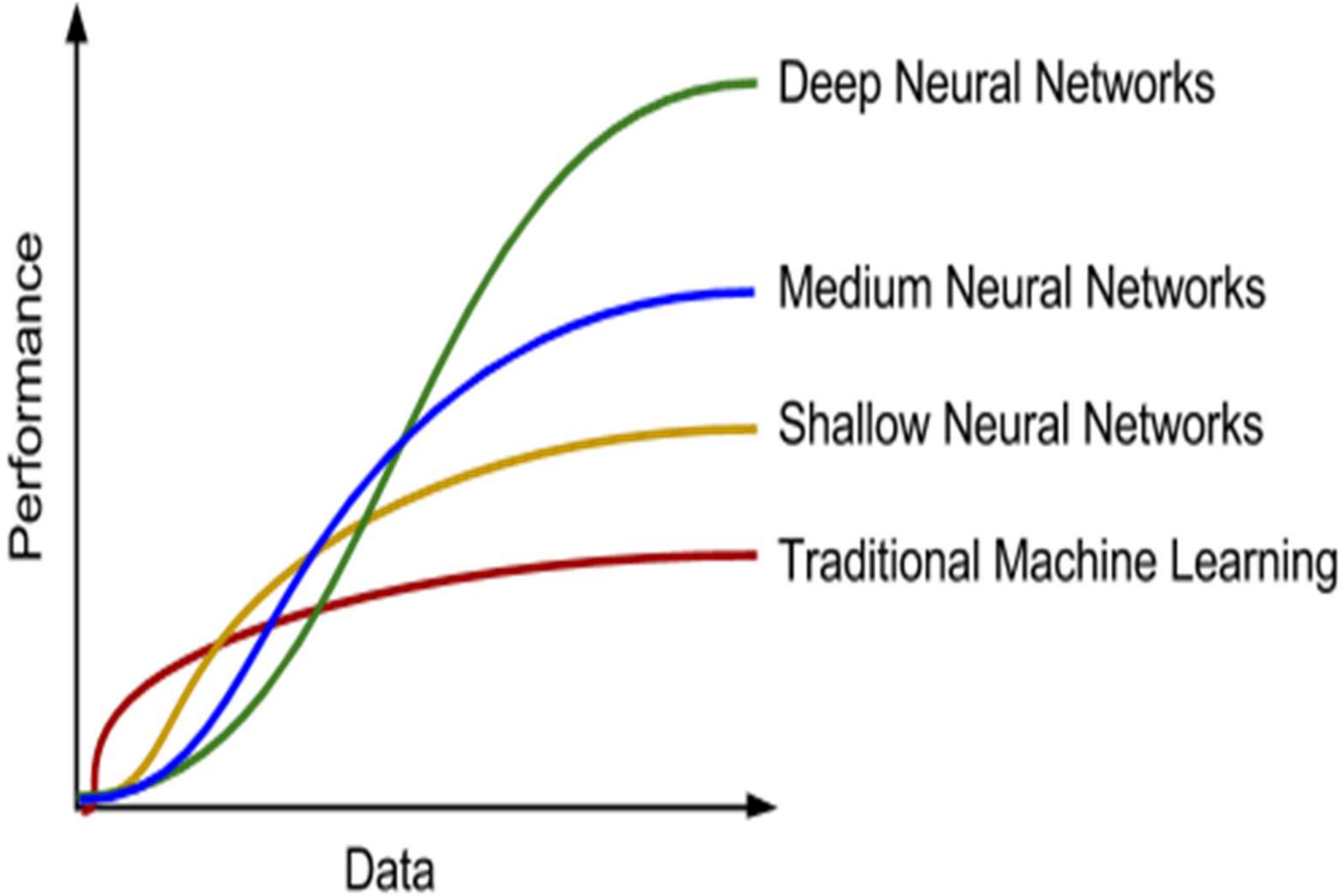
Optimization

ADAM: a method for stochastic optimization
[Kingma et al. 2015]



- https://arxiv.org/pdf/1412.6980.pdf

Data vs. Neural Nets



Deep Learning