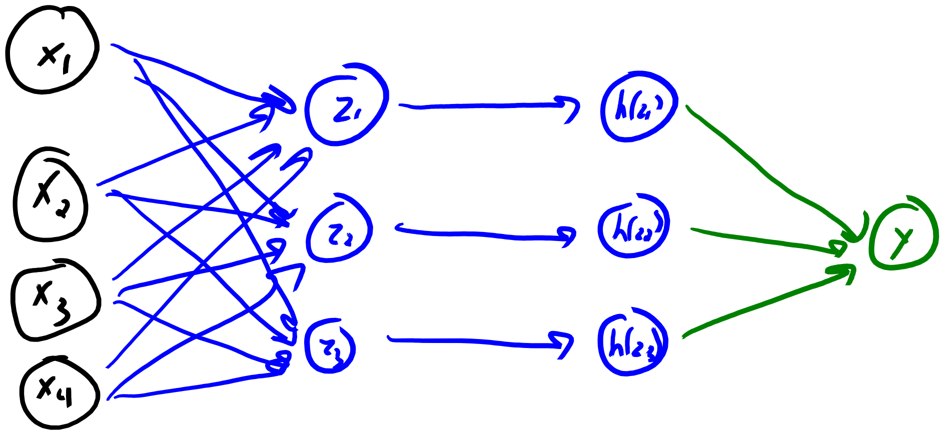
CPSC 440: Machine Learning

Double Descent Curves

Winter 2022

Last Time: Neural Networks

- We discussed neural networks with one hidden layer:



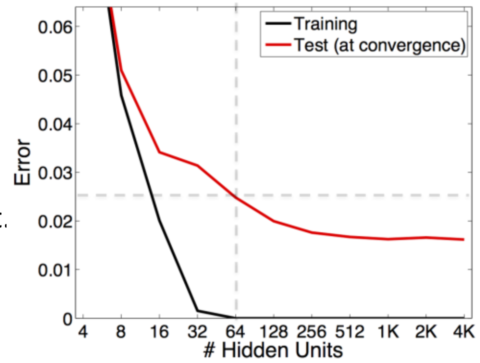
$$\hat{y} = \underbrace{v}_{k\times 1}^{T}\, h(\underbrace{W}_{k\times d}\ \underbrace{x}_{d\times 1})$$

$h: \mathbb{R}^k \mapsto \mathbb{R}^k$ (must be non-linear)

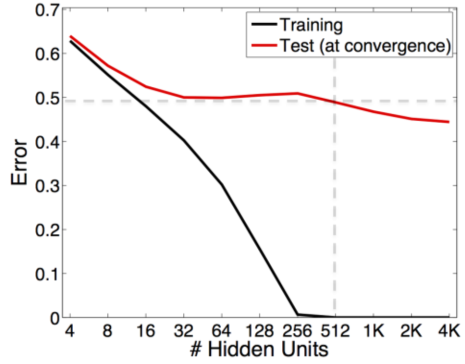
Cost: $O(kd)$

  - "Simultaneously learn the features and the linear model."
  - Often perform better with bias variables and/or residual/skip connections.
  - They are universal approximators (but not the only ones).
  - Leads to non-convex training objective, which we apply SGD to.

MNIST



CIFAR-10

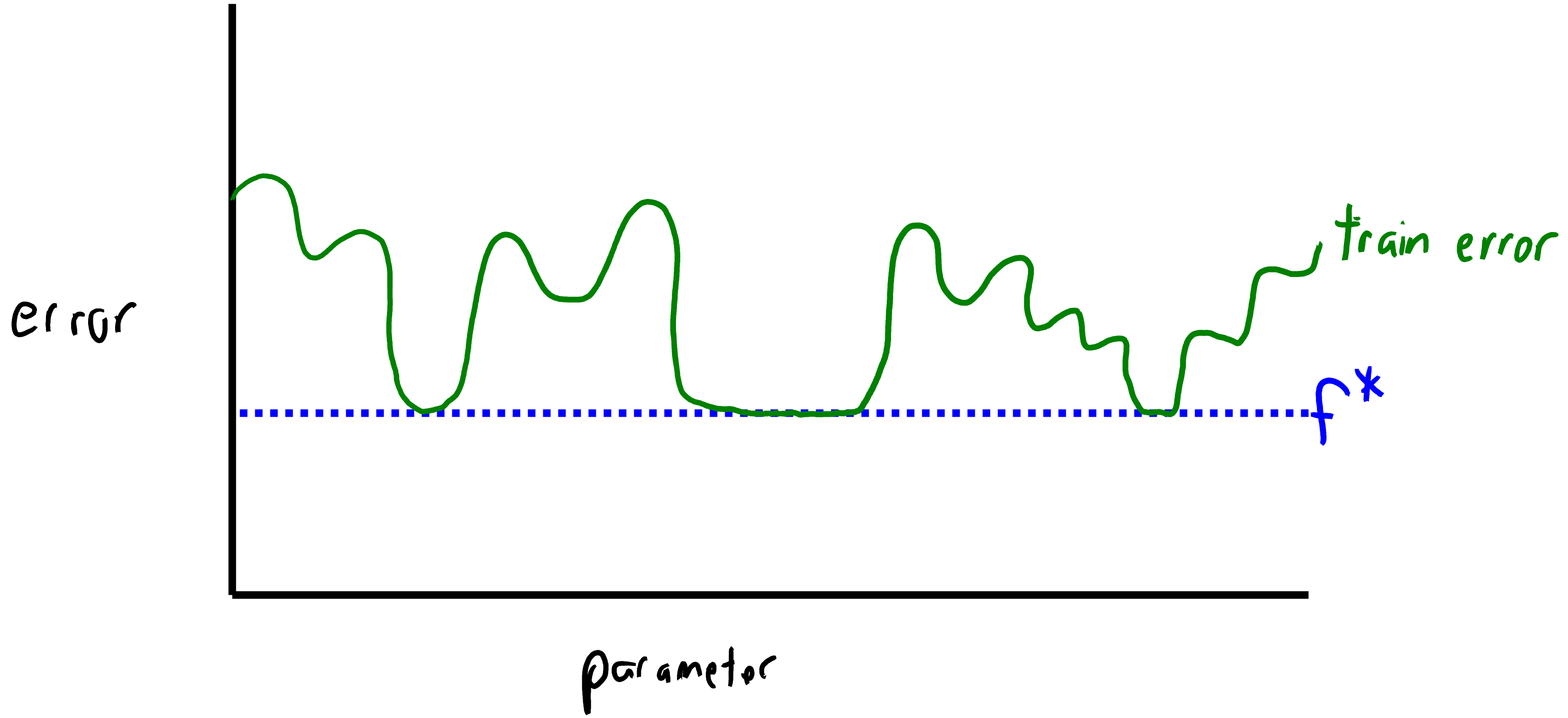


  - Recent experimental observations:
    - With enough hidden units, SGD often finds aa global minimum.
      - Even though training is NP-hard in general.
    - And the global minima it fits does not overfit as much as we expect.

https://www.neyshabur.net/papers/inductive_bias_poster.pdf

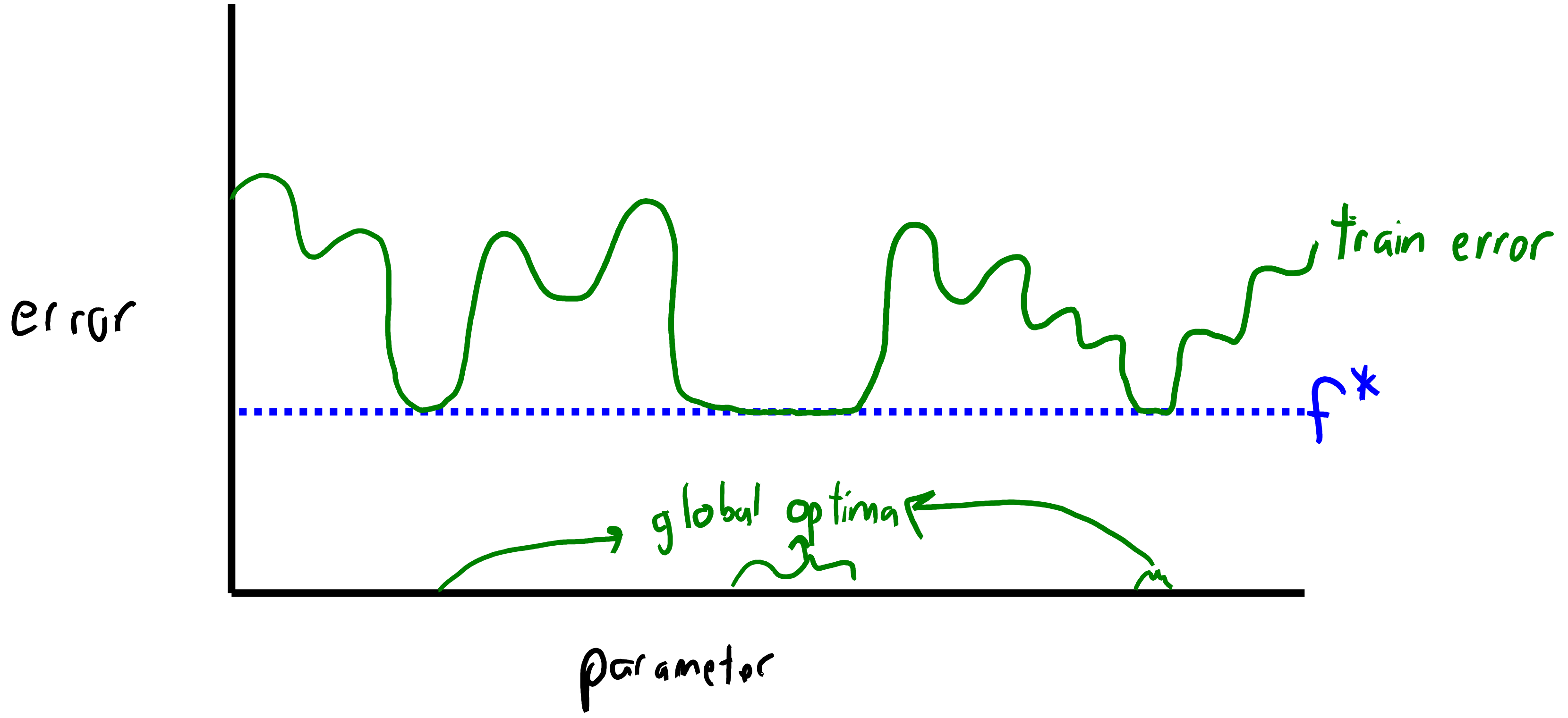
Multiple Global Minima?

- For standard objectives, there is a global min function value f*:



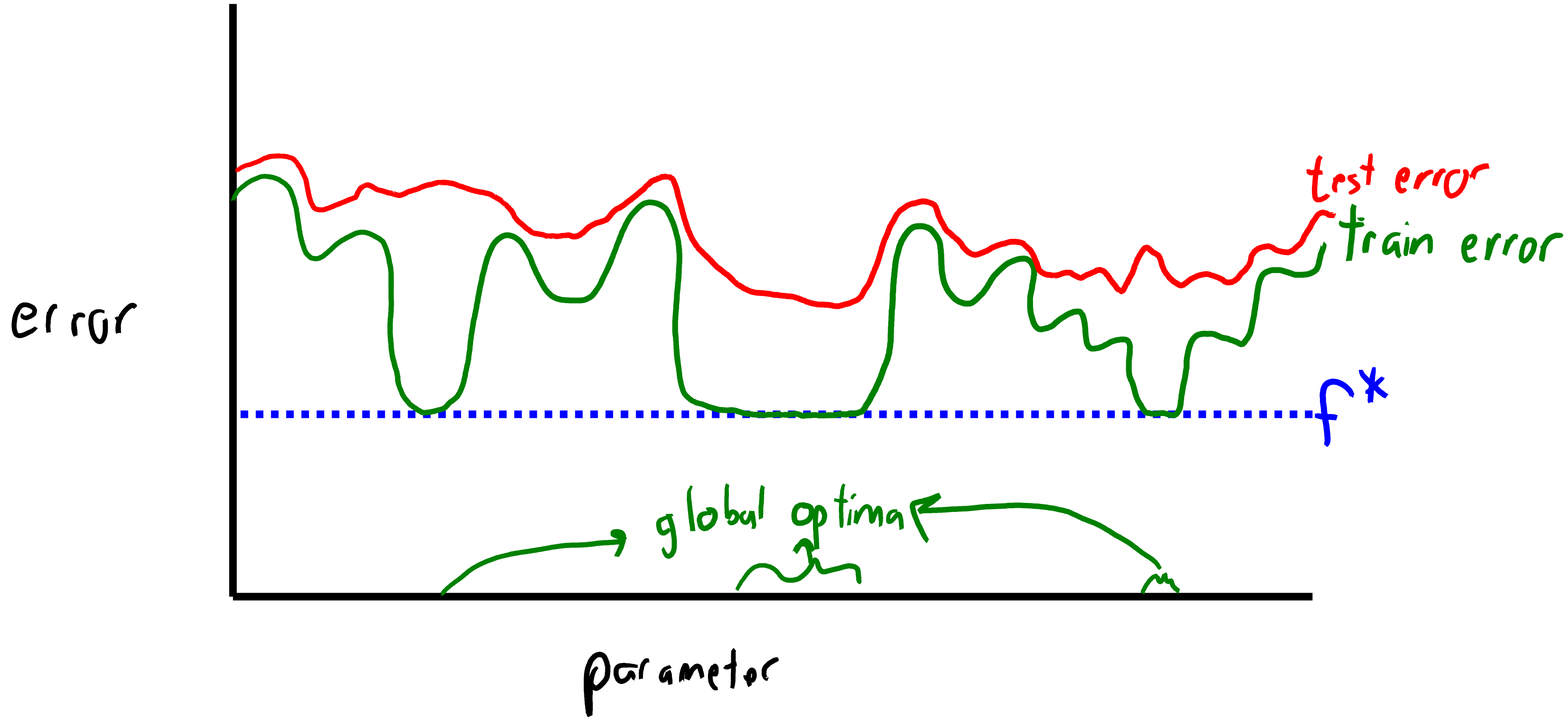
Multiple Global Minima?

- For standard objectives, there is a global min function value f*:



- But this may be achieved by many different parameter values.

Multiple Global Minima?



- These training error "global minima" may have very-different test errors.
- Some of these global minima may be more "regularized" than others.

Implicit Regularization of SGD

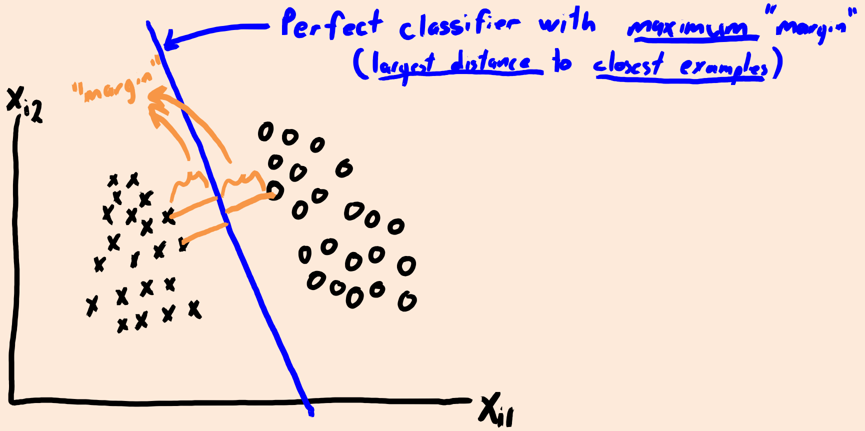
- There is empirical evidence that using SGD regularizes parameters.
  - We call this the "implicit regularization" of the optimization algorithm.

- Beyond empirical evidence, we know this happens in simpler cases.

- Example of implicit regularization:
  - Consider a least squares problem where there exists a 'w' where Xw=y.
    - Residuals are all zero, we fit the data exactly.
  - You run [stochastic] gradient descent starting from w=0.
  - Converges to solution Xw=y that has the minimum L2-norm.
    - So using SGD is equivalent to L2-regularization here, but regularization is "implicit".
    - Using w=X\y in Julia also gives you this regularized solution.

Implicit Regularization of SGD

- Example of implicit regularization:
  - Consider a logistic regression problem where data is linearly separable.
    - A linear model can perfectly separate the data.
  - You run gradient descent from any starting point.
  - Converges to max-margin solution of the problem (minimum L2-norm solution).
    - So using gradient descent is equivalent to encouraging large margin.



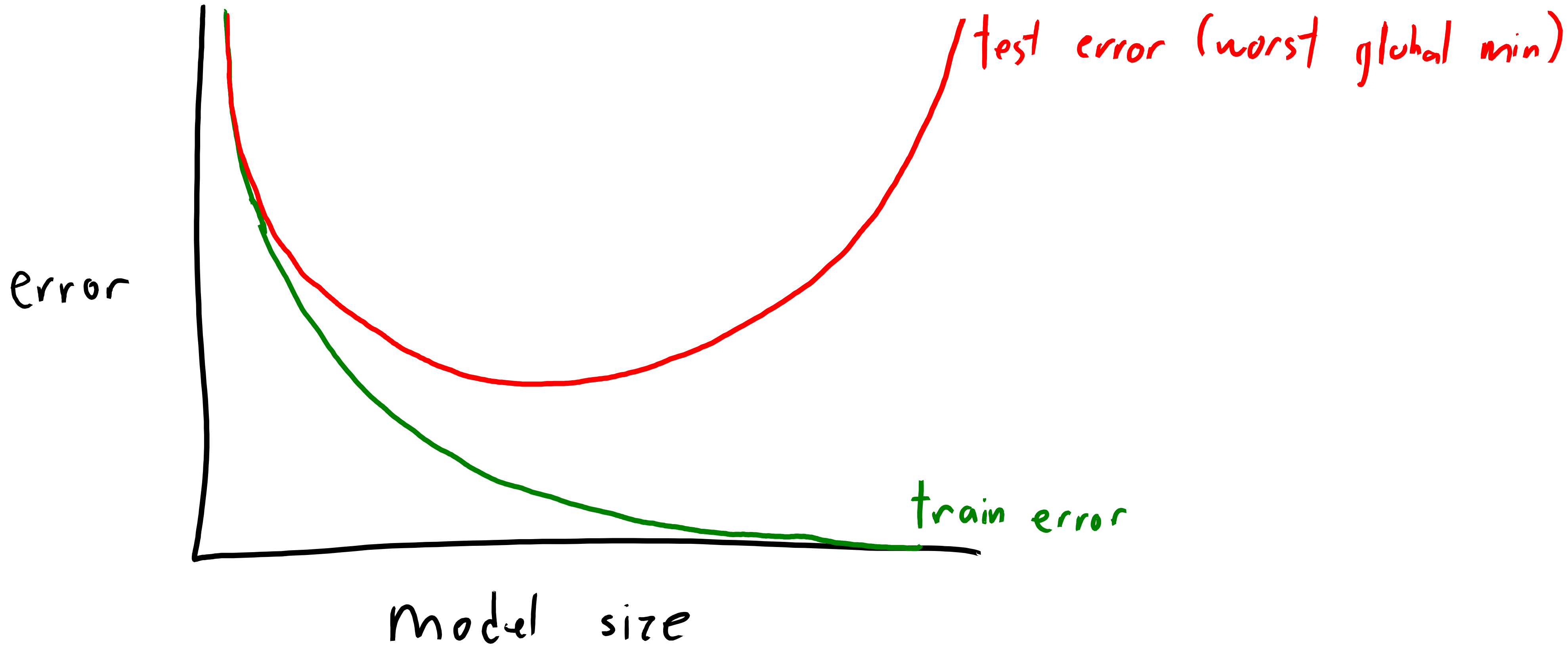
- Related result are known for boosting, matrix factorization, and linear neural networks.

Double Descent Curves

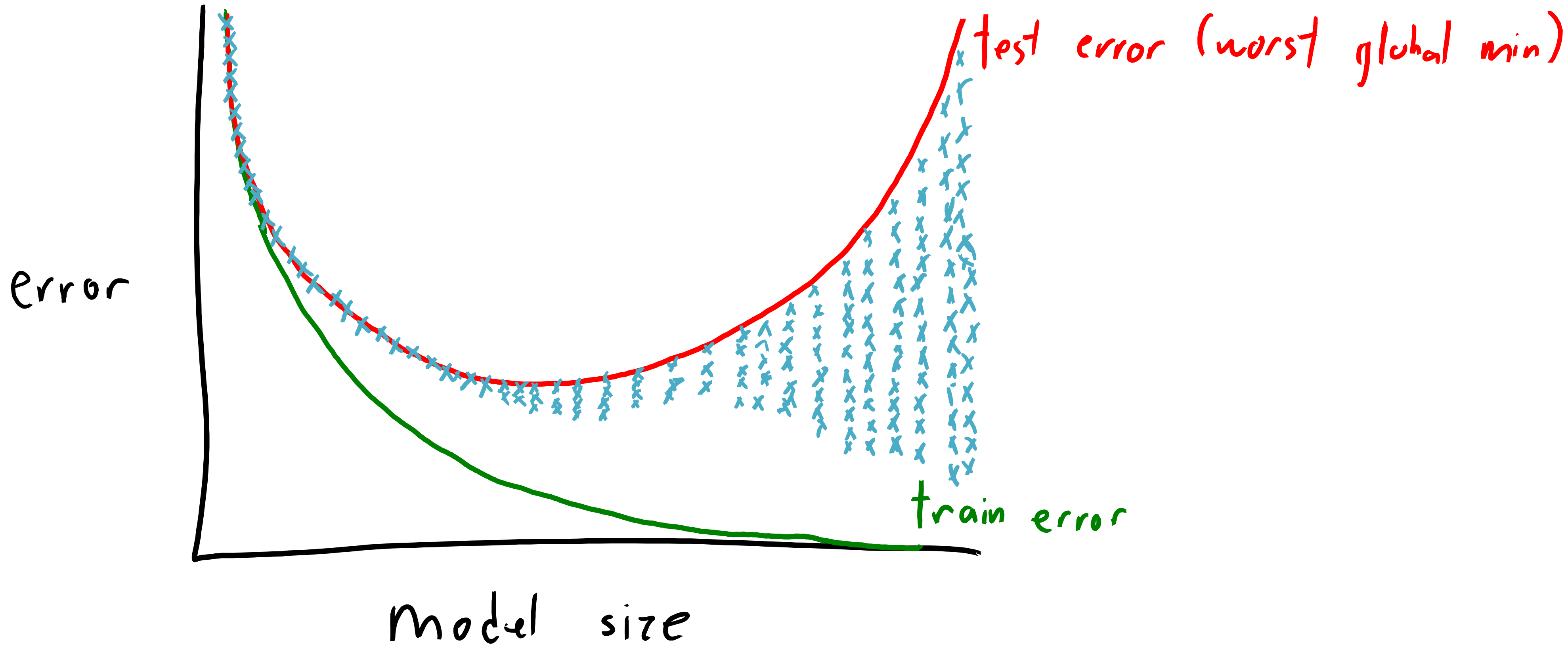
- What is going on???

https://openai.com/blog/deep-double-descent/

Worst vs. Best "Global Minimum"

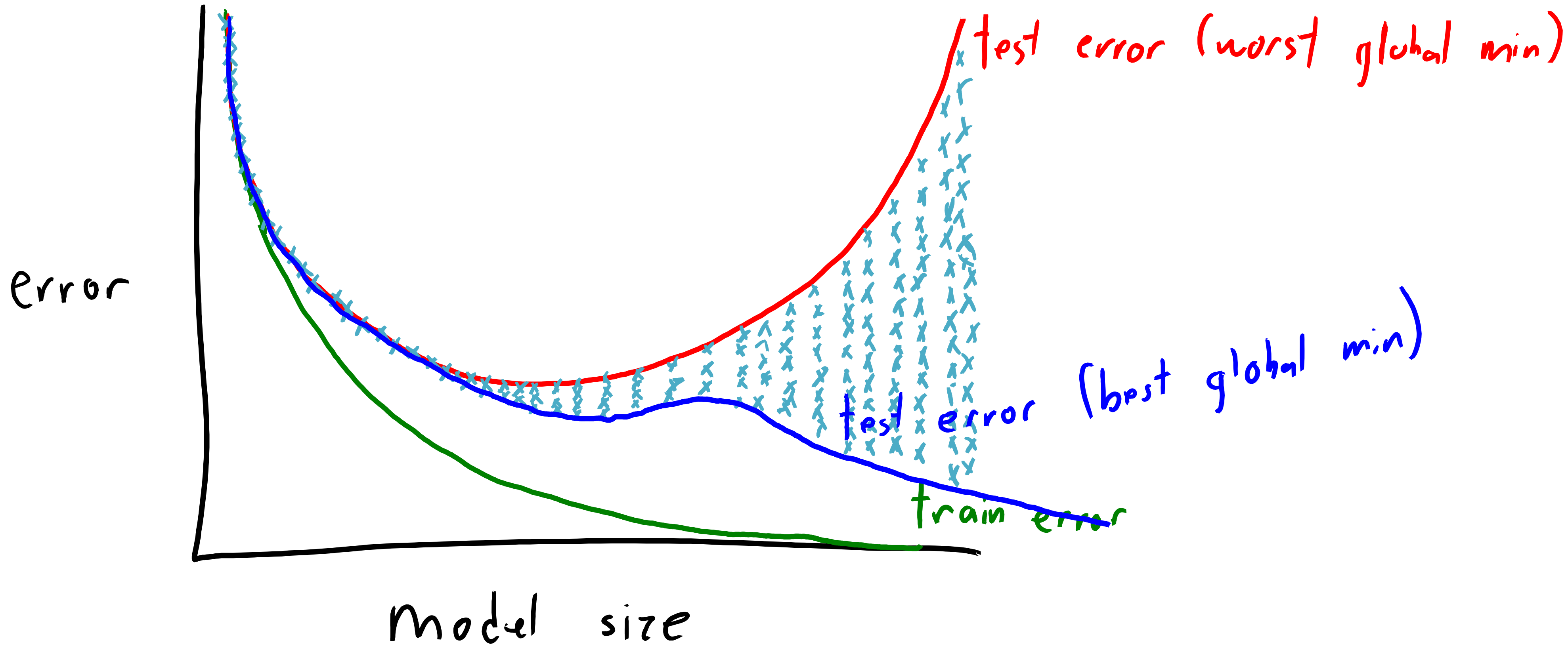


Worst vs. Best "Global Minimum"



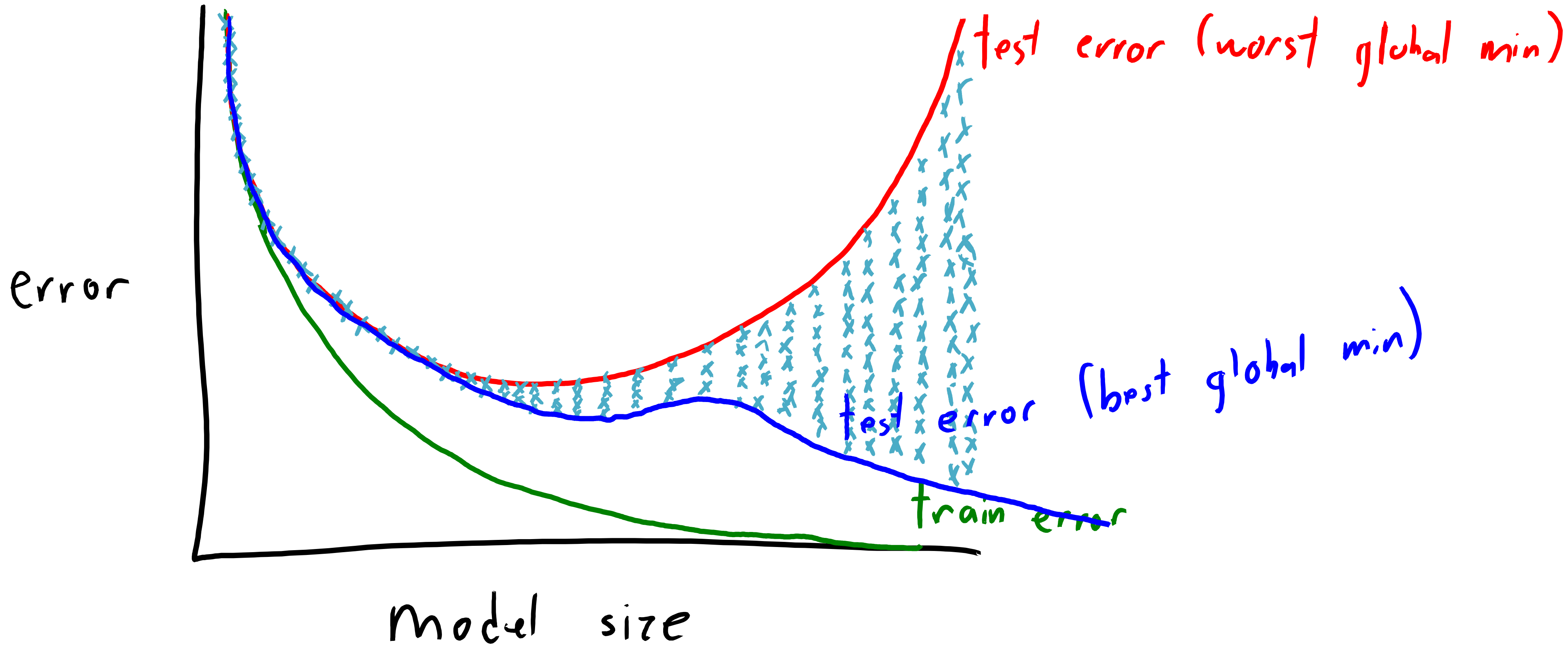
- Learning theory results analyze global min with worst test error.
  - Actual test error for different global minima will be better than worst case bound.
  - Theory is correct, but maybe "worst overfitting possible" is too pessimistic?

Worst vs. Best "Global Minimum"



- Consider instead the global min with best test error.
  - With small models, "minimize training error" leads to unique (or similar) global mins.
  - With larger models, there is a lot of flexibility in the space of global mins (gap between best/worst).
- Gap between "worst" and "best" global min can grow with model complexity.

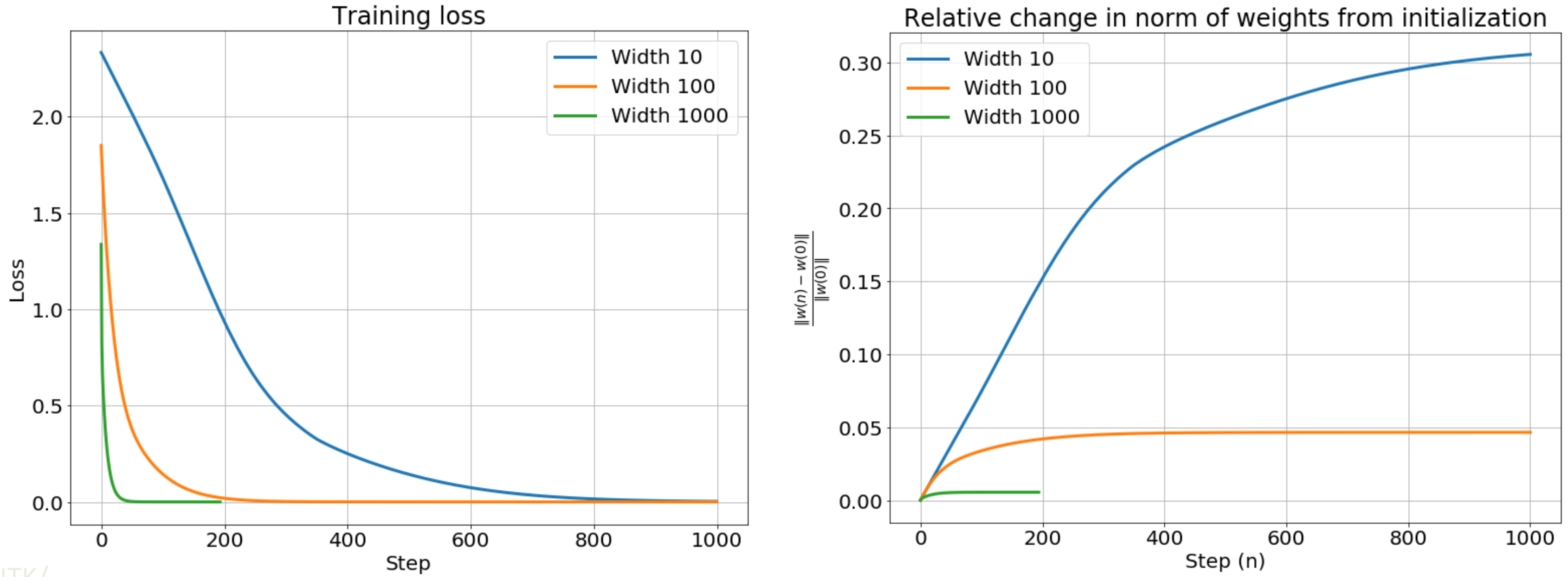
Worst vs. Best "Global Minimum"



- Can get "double descent" curve in practice if parameters roughly track "best" global min shape.
  - One way to do this: increase regularization as you increase model size.
- Maybe "neural network trained with SGD" has "more implicit regularization for bigger models"?
  - But this behavior is not specific to implicit regularization of SGD and not specific to neural networks.

Implicit Regularization of SGD (as function of size)

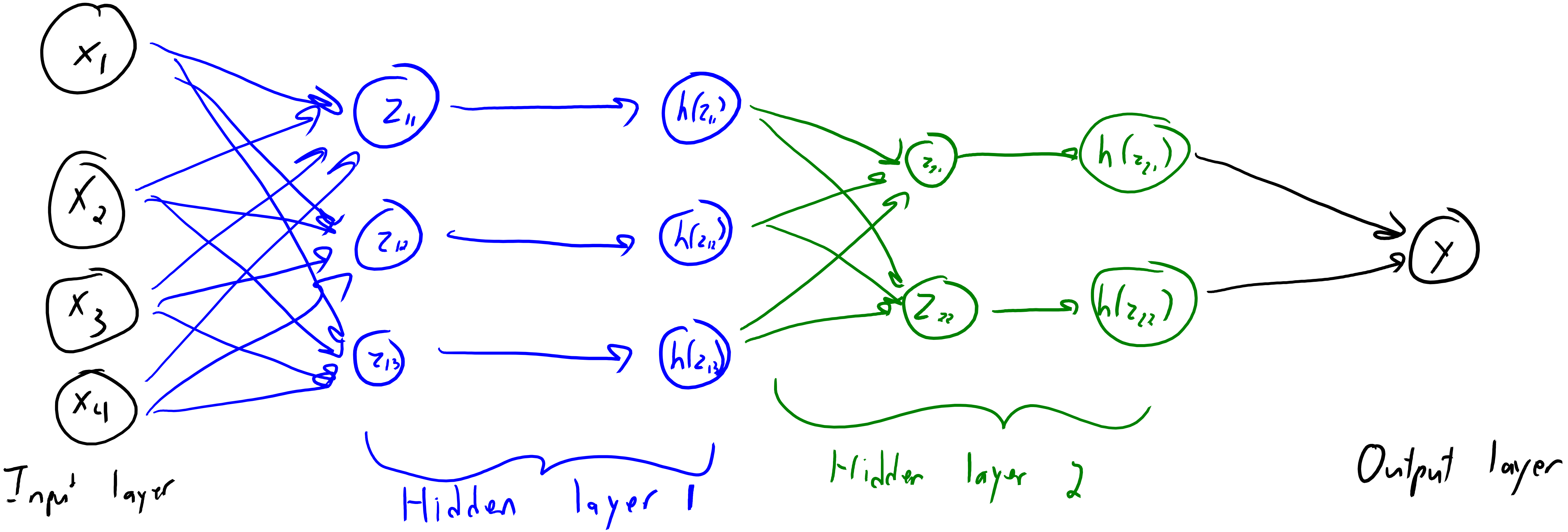
- Why would implicit regularization of SGD increase with dimension?
  - Maybe SGD finds low-norm solutions?
    - In higher-dimensions, there is flexibility in global mins to have a low norm?
  - Maybe SGD stays closer to starting point as we increase dimension?
    - This would be more like a regularizer of the form $||w - w^0||$.



Next Topic: Deep Learning

Deep Learning

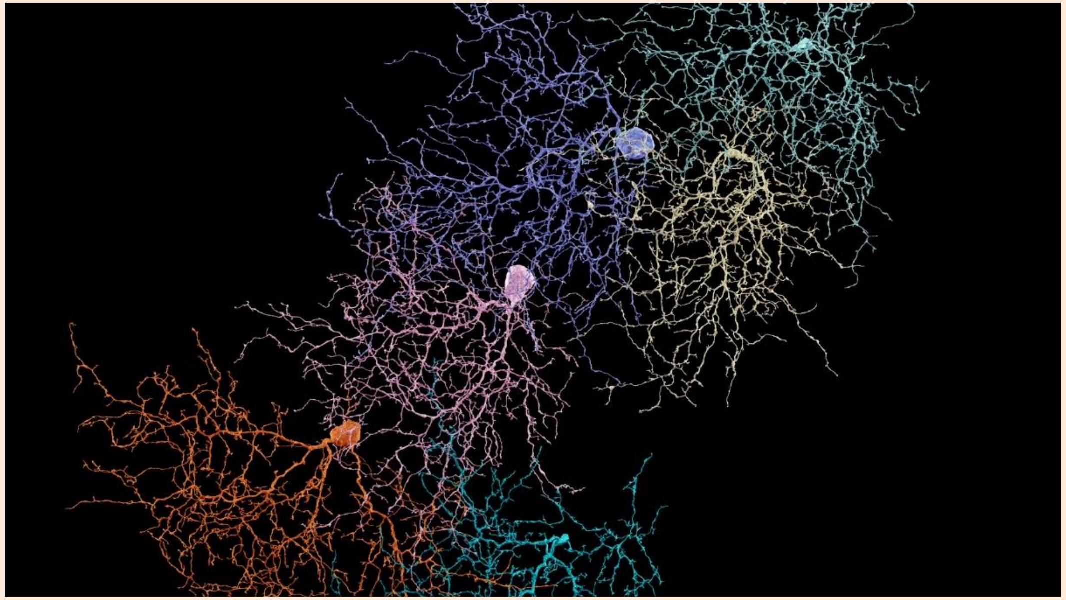
- Deep learning models have more than one hidden layer:



- We transform our activations one or more times.

Why Multiple Layers?

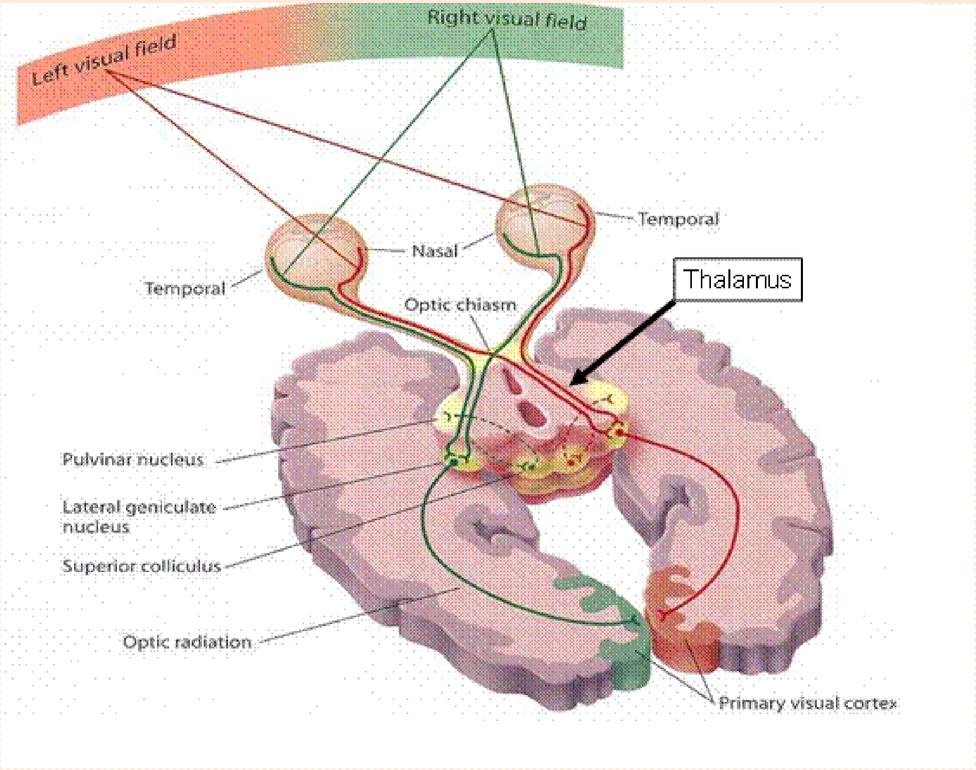
- Historically, deep learning was motivated by "connectionist" ideas:
  - Brain consists of network of highly-connected simple units.
    - Same units repeated in various places.
    - Computations are done in parallel.
    - Information is stored in distributed way.
    - Learning comes from updating of connection strengths.
    - One learning algorithm used everywhere.

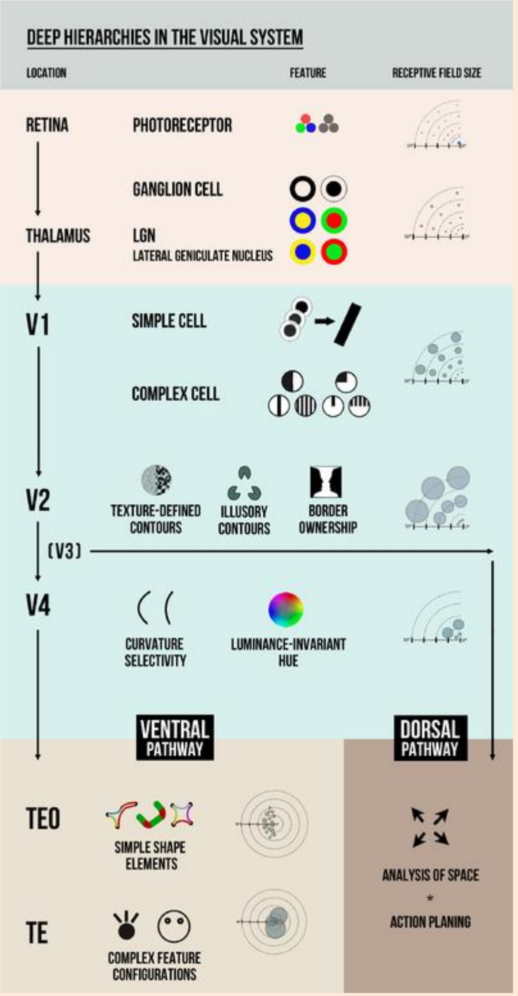


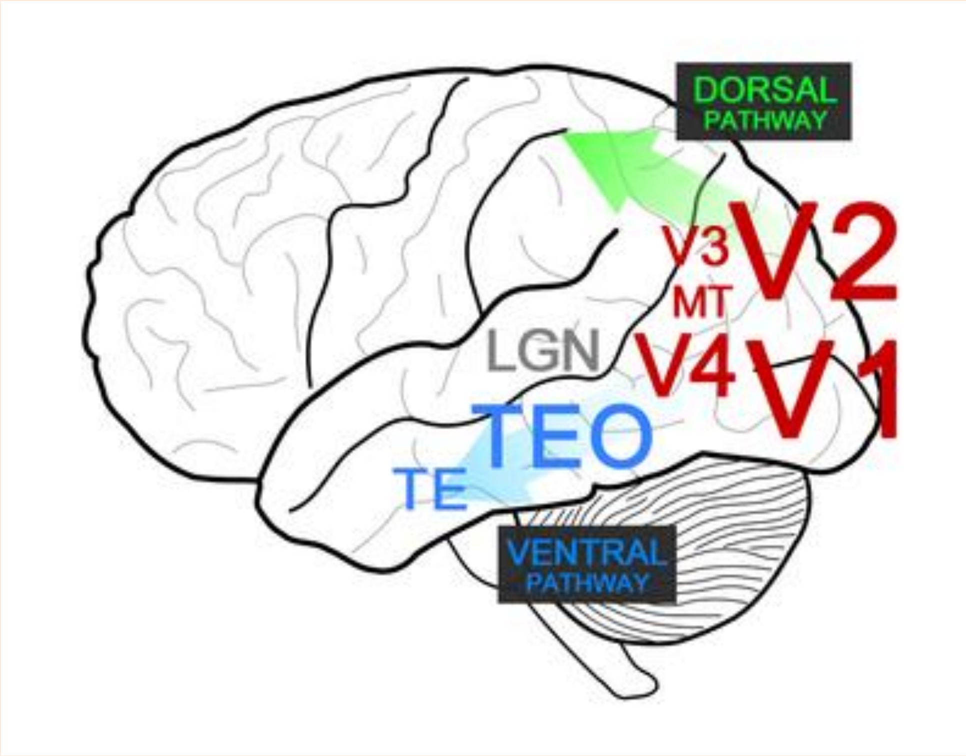
https://www.nytimes.com/2015/01/11/magazine/sebastian-seungs-quest-to-map-the-human-brain.html

Why Multiple Layers?

- And theories on the hierarchical organization of the visual system:



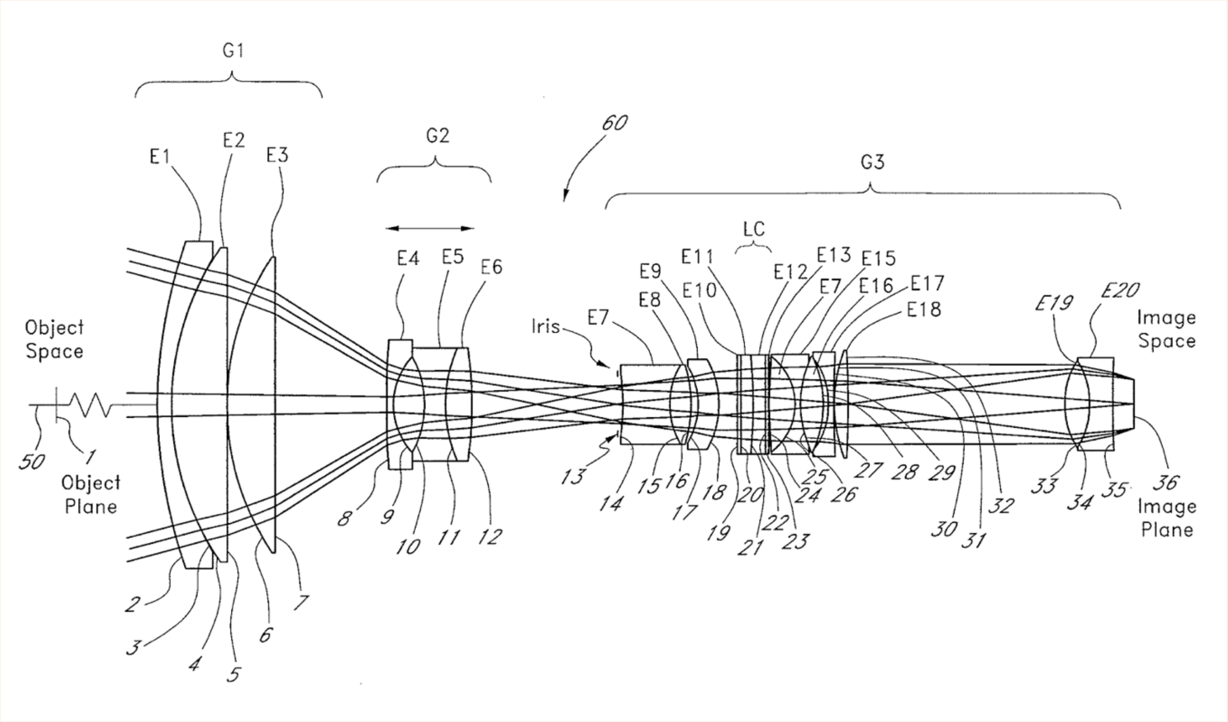


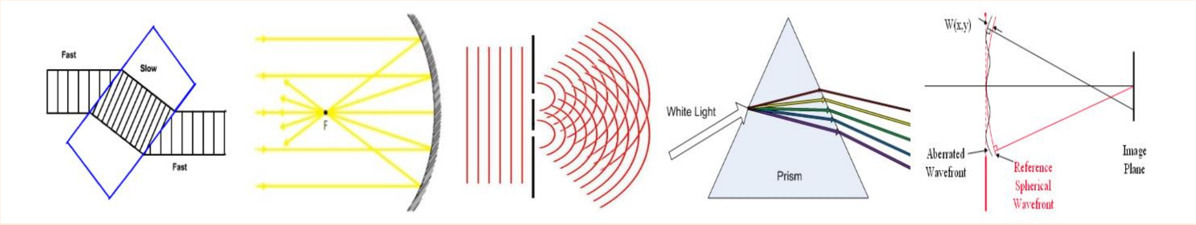


http://www.strokenetwork.org/newsletter/articles/vision.htm
https://en.wikibooks.org/wiki/Sensory_Systems/Visual_Signal_Processing

Why Multiple Layers?

- The idea of multi-layer designs appears in engineering too:
  - Deep hierarchies in camera design:

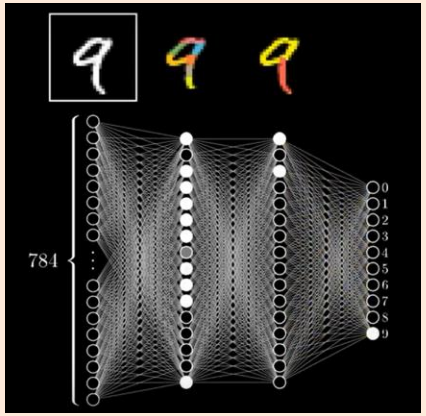




http://www.argmin.net/2018/01/25/optics/

Why Multiple Layers?

- There are also mathematical motivations for using multiple layers:
  - 1 layer gives us a universal approximator of any (reasonable) function.
    - But this layer might need to be huge.
  - With deep networks:
    - Some functions can be approximated with exponentially-fewer parameters.
      - Compared to a network with 1 hidden layer.
    - So deep networks may need fewer parameters than "shallow but wide" networks.
      - And hence may need less data to train.
- Watch this video:
  - https://www.youtube.com/watch?v=aircAruvnKk



Inference In Deep Neural Networks

- The "textbook" choice for deep neural networks:
  - Alternate between doing linear transformations and non-linear transforms.

$$\hat{y} = v^T h(W^4 h(W^3 h(W^2 h(W^1 x))))$$

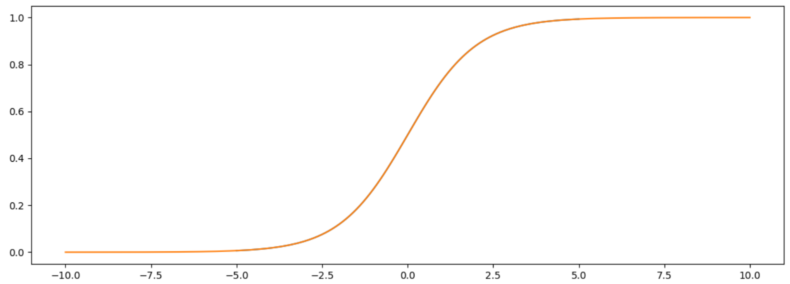
  - Each "layer" might have a different size.
    - W^1 is $k^1 \times d$.
    - W^2 is $k^2 \times k^1$.
    - W^3 is $k^3 \times k^2$.
    - W^4 is $k^4 \times k^3$.
    - v is $k^4 \times 1$.

```
z[1] = W1*x
for layer in 2:nLayers
    z[layer] = Wm[layer-1]*h(z[layer-1])
end
yhat = v'*h(z[end])
```

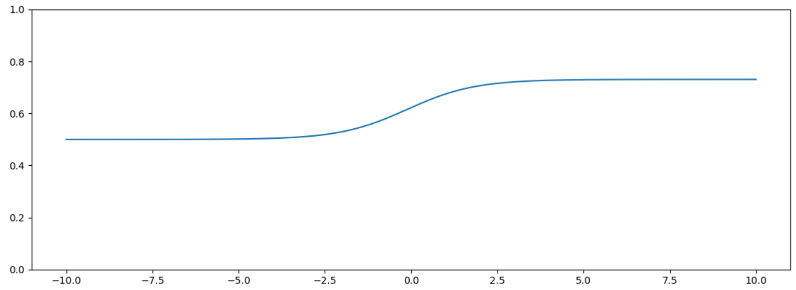
  - We use the same non-linear transform, such as sigmoid, at each layer.
  - Cost for prediction, which is called "forward propagation":
    - Cost of the matrix multiplies: $O(k^1 d + k^2 k^1 + k^3 k^2 + k^4 k^3)$
    - Cost of the non-linear transforms is $O(k^1 + k^2 + k^3 + k^4)$, so does not change cost.
  - Once you have $\hat{y}$, inference works as it does for Bernoulli with $\theta = 1/(1+\exp(-\hat{y}))$.

New Issue: Vanishing Gradients

- Consider the sigmoid function:



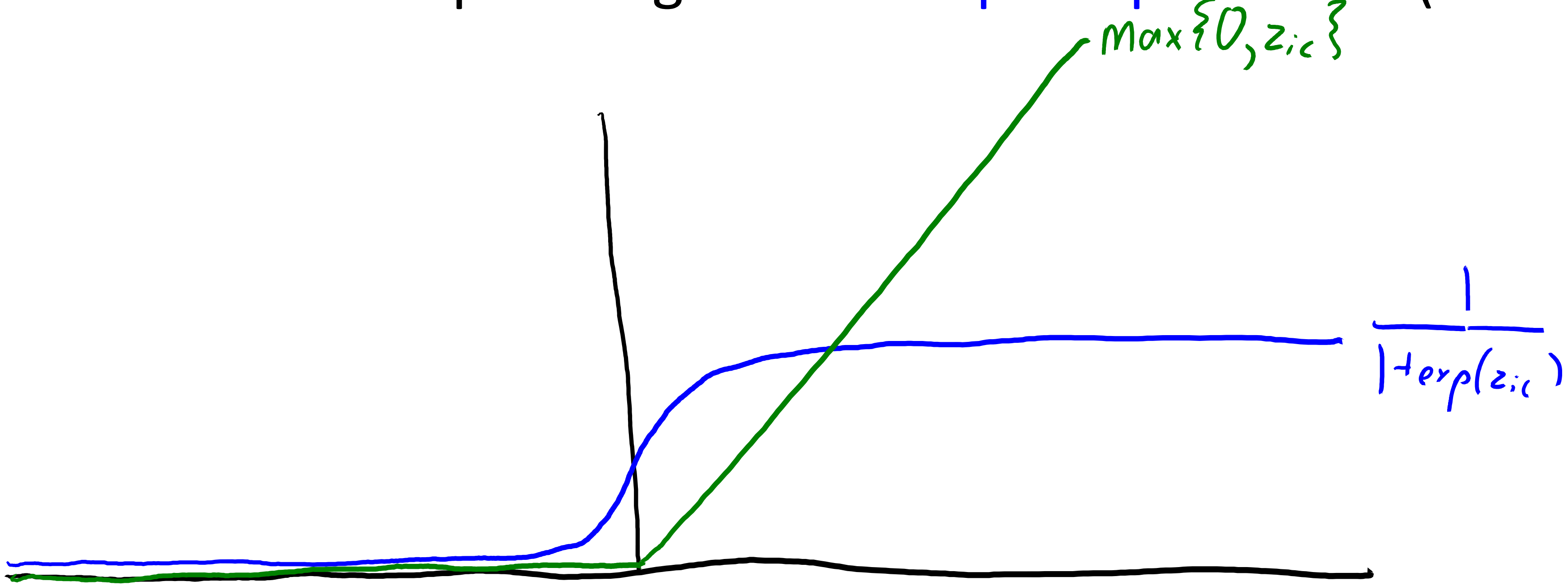
- Away from the origin, the gradient is nearly zero.
- The problem gets worse when you take the sigmoid of a sigmoid:



- In deep networks, many gradients can be nearly zero everywhere.
  - And numerically they will be set to 0.

Rectified Linear Units (ReLU)

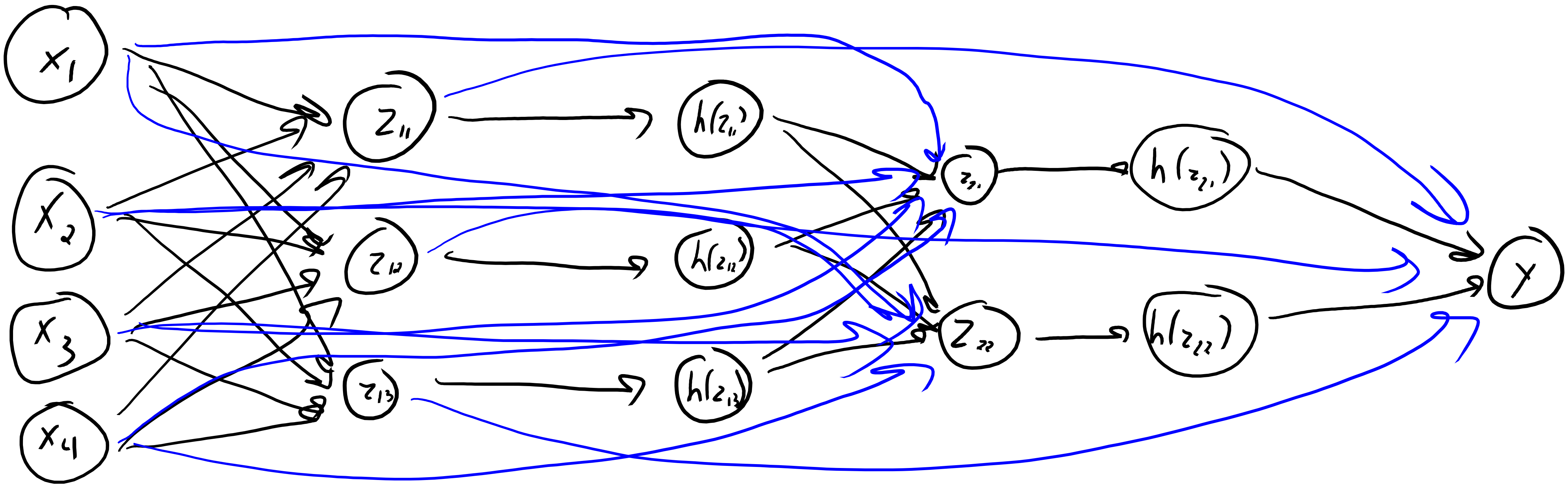
- Modern networks often replace sigmoid with perceptron loss (ReLU):



- Just sets negative values z_{ic} to zero.
  - Reduces vanishing gradient problem (positive region is never flat).
  - Gives sparser activations.
  - Still gives a universal approximator if size of hidden layers grows with ‘n’.

Skip Connections Deep Learning

- Skip connections can also reduce vanishing gradient problem:



- Makes “shortcuts” from input to output with fewer transformations.
  - Many variations exist on skip connections locations and how they are used.

Summary

- Implicit regularization and double descent curves.
  - Possible explanations for why deep networks often generalize well.
- Deep learning:
  - Neural networks with multiple hidden layers.
  - Can allow learning with smaller models and less data than “wide” networks.
- Vanishing gradient in deep networks (gradient may be close to 0).
  - Can be reduced using rectified linear units (ReLU) as non-linear transform.
  - Can be reduced using various forms of skip connections.

- Next time: how to avoid writing nasty derivatives by hand.