Learning Hyperparameters

Andrew Saxe

neuromatch academy

The effect of depth on training

Now that we can implement a network, let's understand some core learning behaviors and tradeoffs

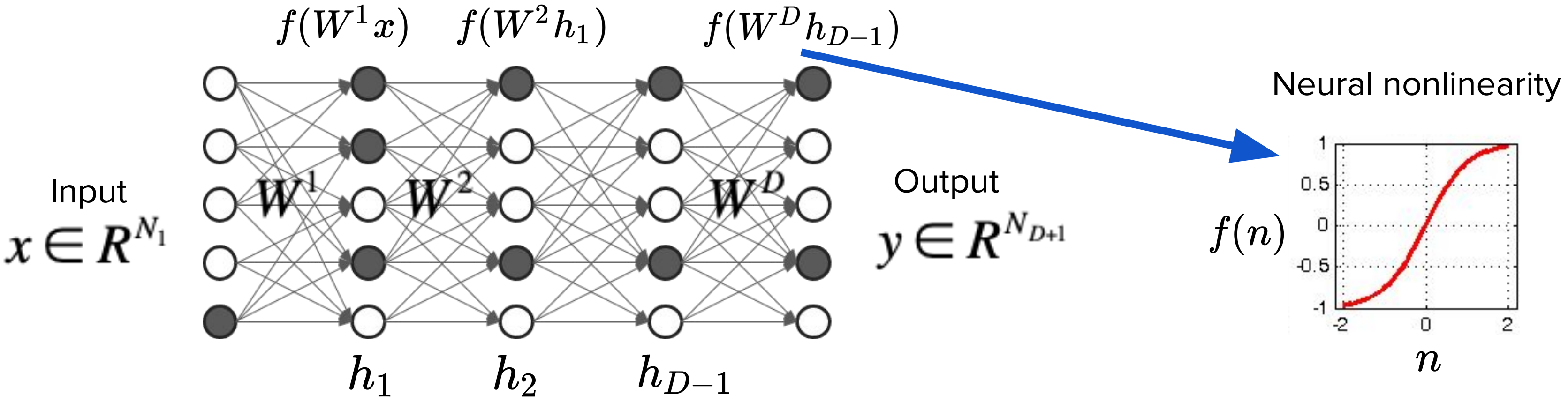
The architecture, initialization, and learning hyperparameters all can change the performance of a network dramatically

To be proficient at training deep networks, we have to build our intuition

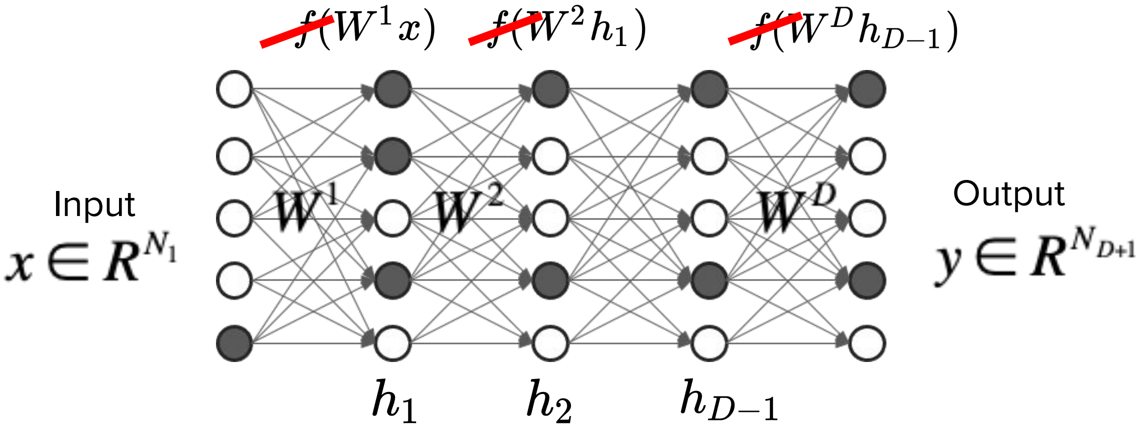
Opening the black box

Deep Network

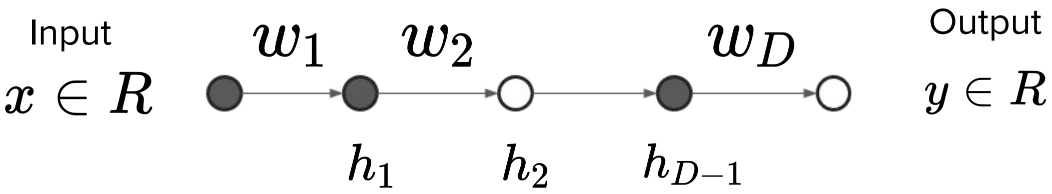
Little hope to understand full modern systems in detail



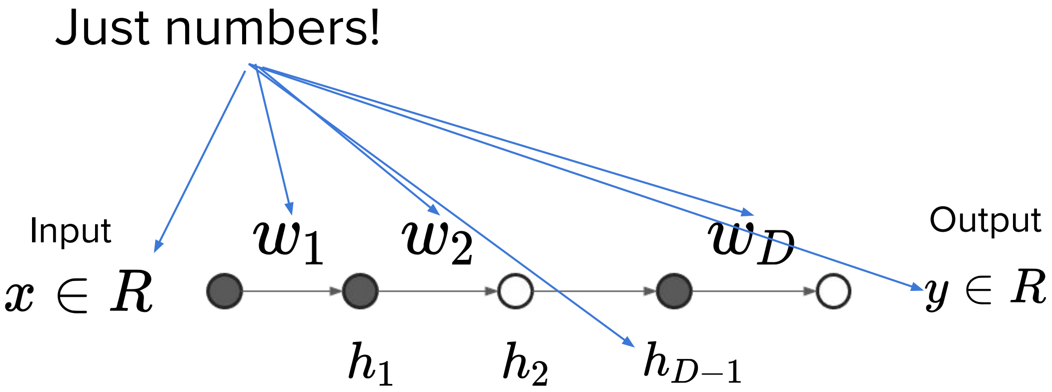
Deep *Linear* Network



Deep *Narrow* Linear Network

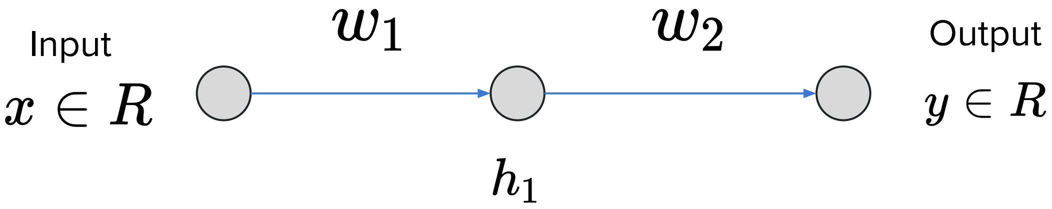


Deep *Narrow* Linear Network



$$y = w_D w_{D-1} \cdots w_1 x$$

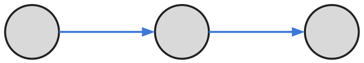
1 Layer Narrow Linear Network



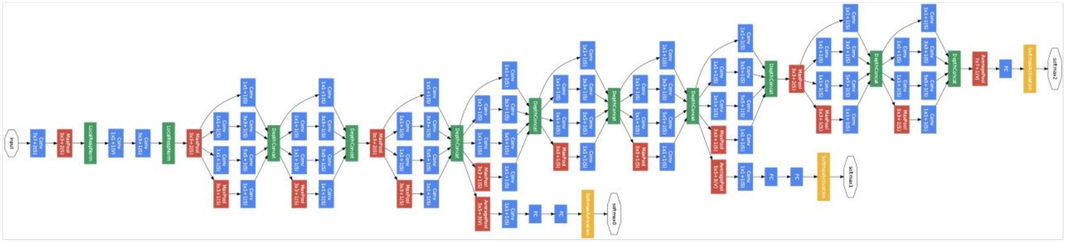
$$y = x w_1 w_2$$

Simple models

Today



The rest of your career

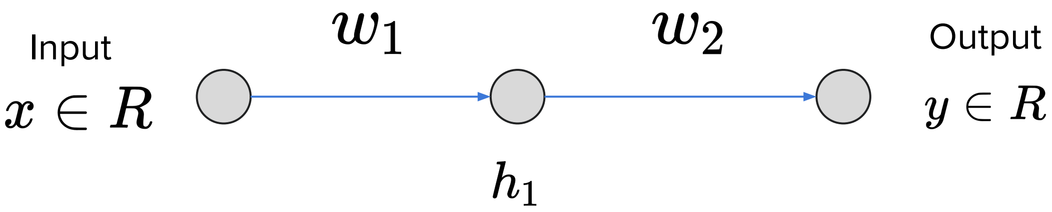


Szegedy et al., CVPR 2015

1 Layer Narrow Linear Network

Dataset: $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_P, y_P)\}$

Mean squared error loss: $L(w_1, w_2) = \frac{1}{P}\sum_{p=1}^{P}(y_p - \hat{y}_p)^2$



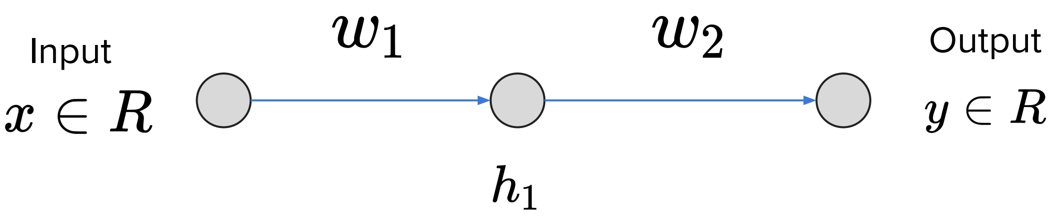
$$y = xw_1w_2$$

Loss from one example $= \frac{1}{P}(y_p - x_p w_1 w_2)^2$

1 Layer Narrow Linear Network

Dataset: $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_P, y_P)\}$

Mean squared error loss: $L(w_1, w_2) = \frac{1}{P} \sum_{p=1}^{P} (y_p - \hat{y}_p)^2$



$$y = x w_1 w_2$$

Implement gradient descent

Training landscape

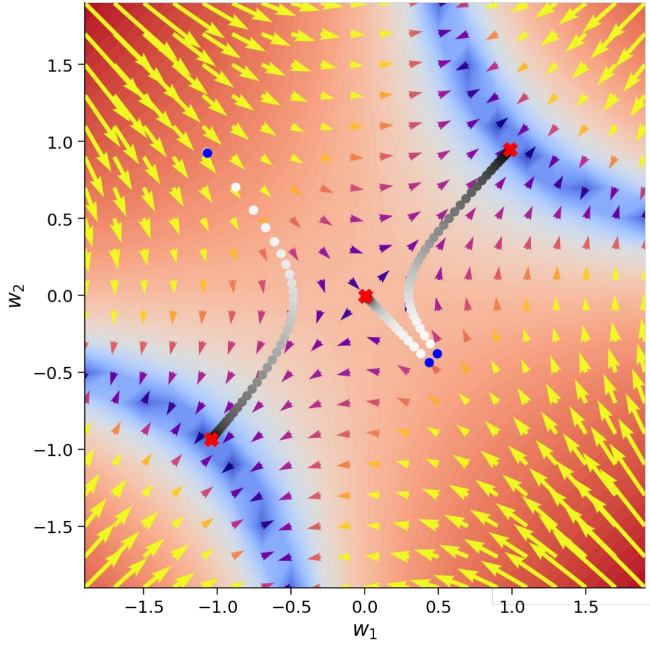
We train networks by minimizing the loss function

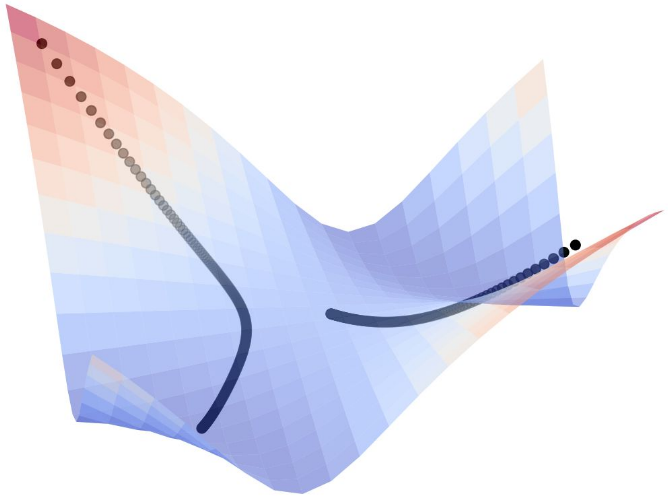
What do these loss landscapes actually look like?

Usually loss landscapes are impossible to plot because they are in high dimensions, but here we can examine it directly

Explore this loss landscape and the resulting GD trajectories

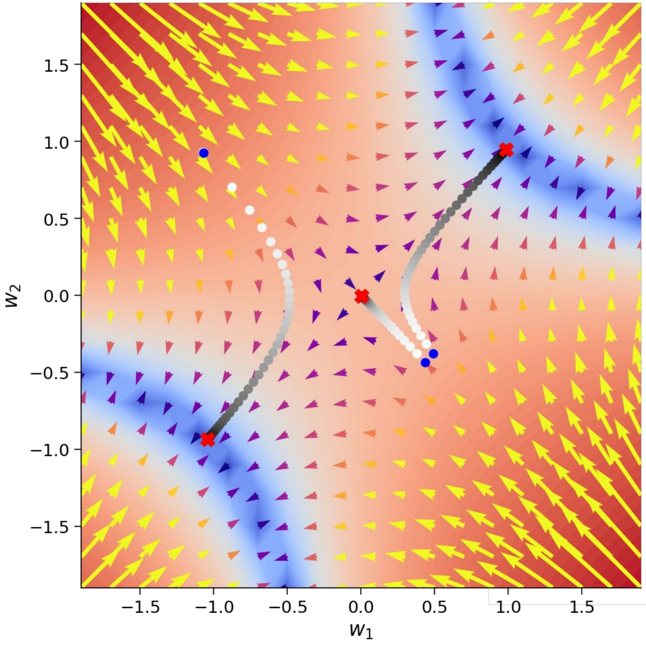
Anatomy of a landscape

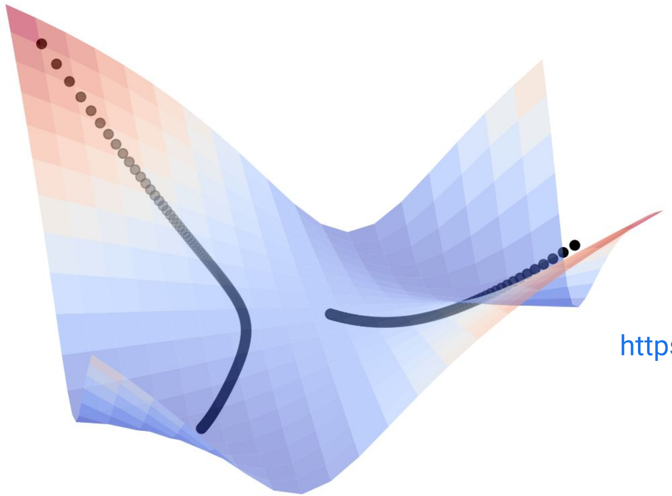




Critical points: where the gradient is zero and dynamics stop
Minimum: surrounding points are not lower
Maximum: surrounding points are not higher
Saddle point: some descent directions, some ascent directions

Anatomy of a landscape



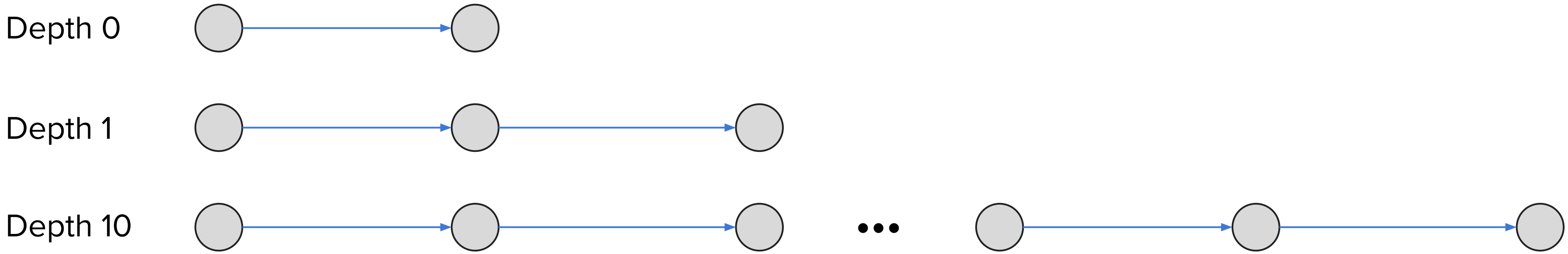


https://losslandscape.com/explorer

Critical points: where the gradient is zero and dynamics stop
Minimum: surrounding points are not lower
Maximum: surrounding points are not higher
Saddle point: some descent directions, some ascent directions

The effect of depth on training

How does network depth impact training speed, everything else being equal?



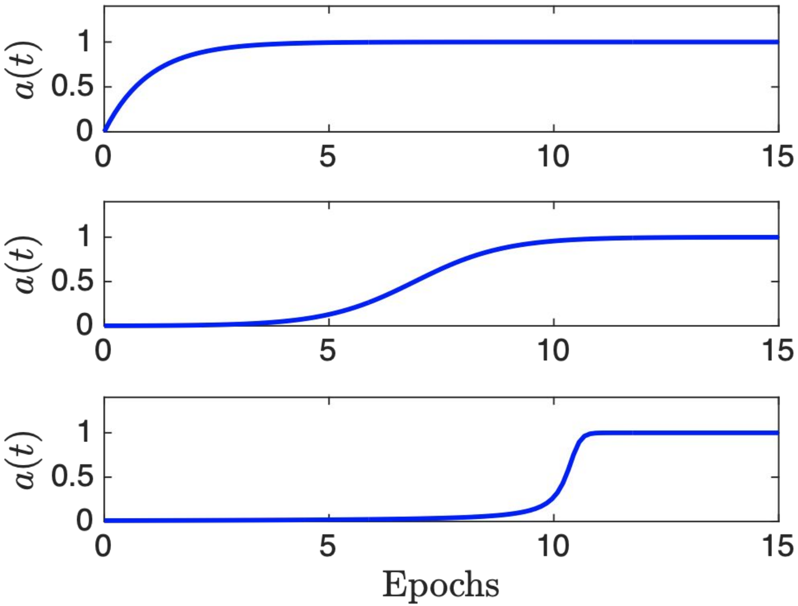
Explore how depth changes learning trajectories

The effect of depth on training

Shallow (D=0):

Deep (D=1):

V. Deep ($D \longrightarrow \infty$):



$$a(t) = w_{D+1} w_D \cdots w_2 w_1$$

Choosing a learning rate

How to pick η? $$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L(\mathbf{w}^{(t)})$$

The gradient points in the steepest descent direction for *infinitesimal* step sizes

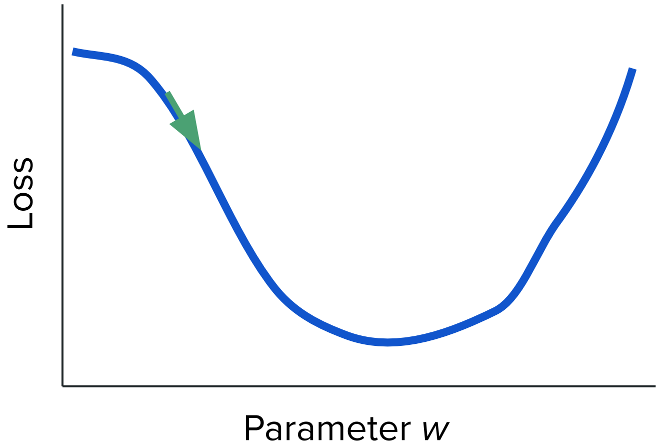
But infinitesimal step sizes don't take you very far!

Play with learning rate. Learn to diagnose issues from error curves.

Choosing a learning rate

How to pick η? $$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L(\mathbf{w}^{(t)})$$

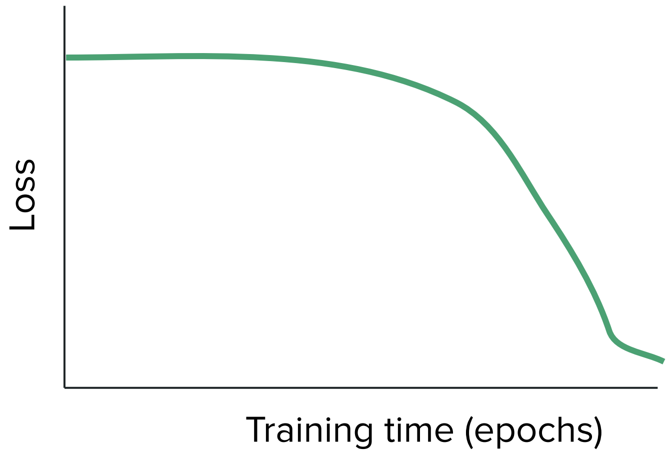
Too small:
flat line

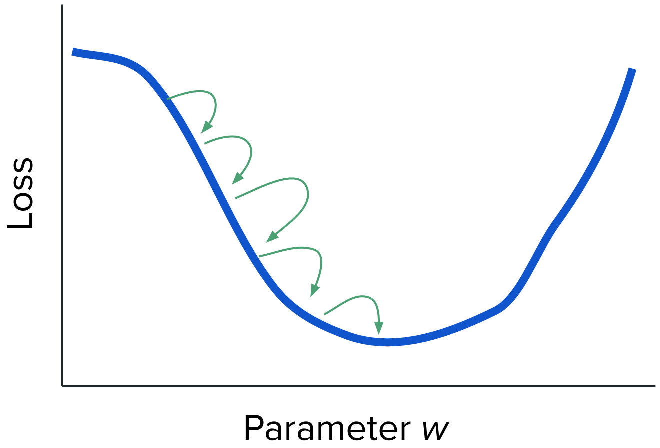


Choosing a learning rate

How to pick η? $$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L(\mathbf{w}^{(t)})$$

Slightly too small: works but slow

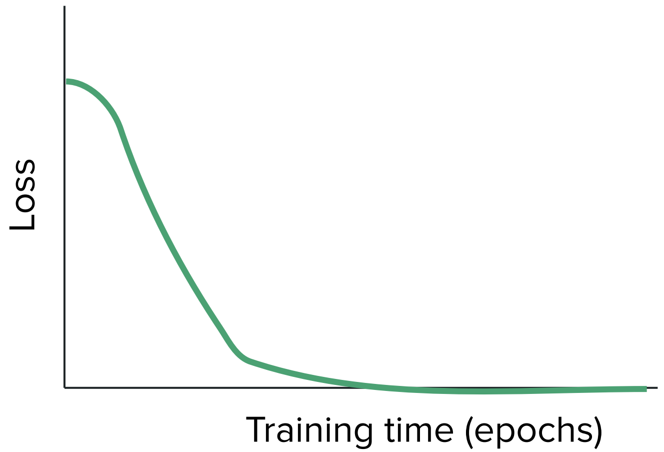


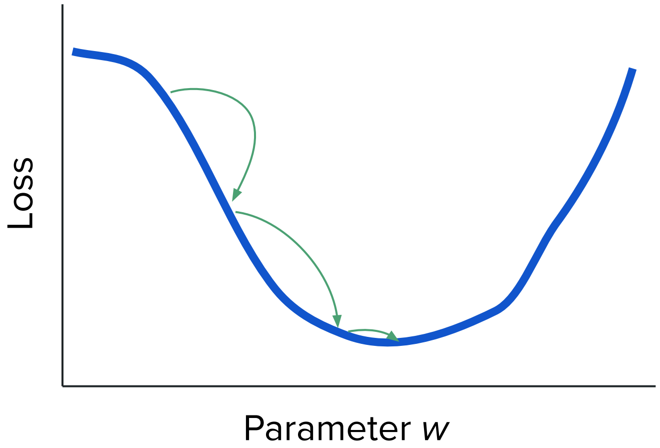


Choosing a learning rate

How to pick η? $$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L(\mathbf{w}^{(t)})$$

Just right: converges quickly and cleanly

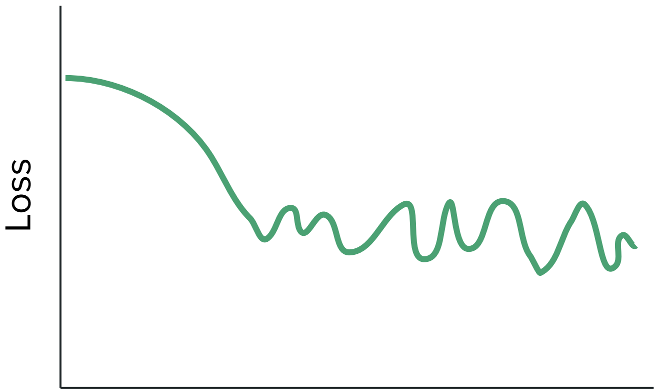


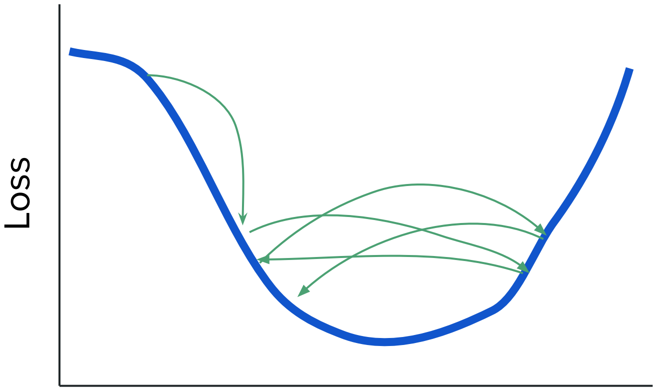


Choosing a learning rate

How to pick η? $$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L(\mathbf{w}^{(t)})$$

Slightly too big: chaotic

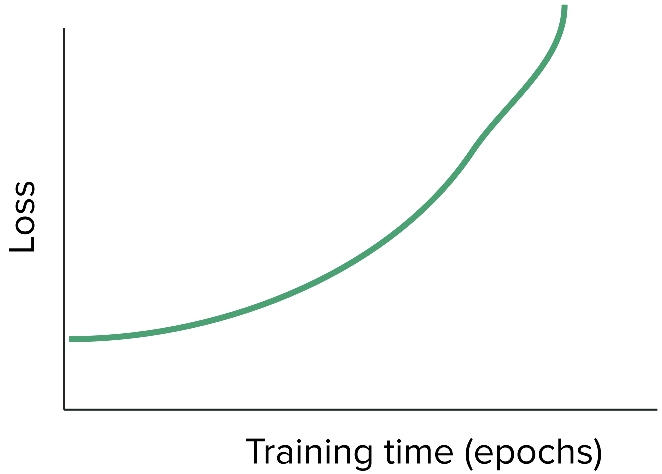


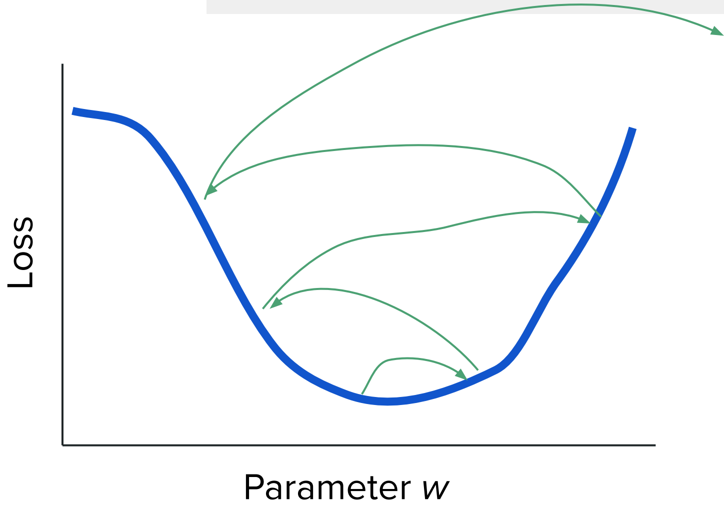


Choosing a learning rate

How to pick η? $$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L(\mathbf{w}^{(t)})$$

Way too big:
Divergence





Choosing a learning rate

How to pick η? $$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla L(\mathbf{w}^{(t)})$$

Lesson for practice: Aim for the maximum stable learning rate

Depth and learning rate

Unfortunately, hyperparameters interact

The right learning rate for one depth may not be the right learning rate for another

Do deeper networks need larger or smaller learning rates? Are deep networks still slower to train if you optimize the learning rate for each?

Play with both depth and learning rate.

Depth and learning rate

Unfortunately, hyperparameters interact

The right learning rate for one depth may not be the right learning rate for another

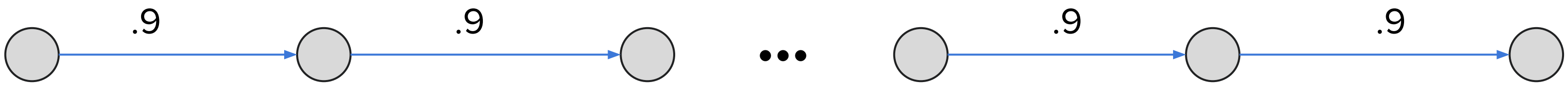
In general, deeper networks need smaller learning rates

Lesson for practice: Carefully optimise all hyperparameters for every architecture you try (this may require many computers :)

Initialisation matters

Unlike in shallow networks, learning in deep networks is exquisitely sensitive to initialisation

Basic reason: products of numbers vanish or explode $y = (\prod_{i=1}^{D} w_i)x$

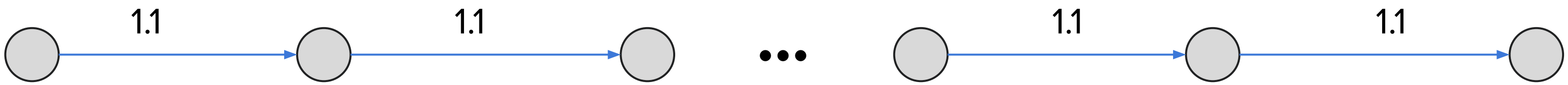


$$y = (0.9)^{100} x = 0.0000265x$$

Initialisation matters

Unlike in shallow networks, learning in deep networks is exquisitely sensitive to initialisation

Basic reason: products of numbers vanish or explode $y = (\prod_{i=1}^{D} w_i)x$

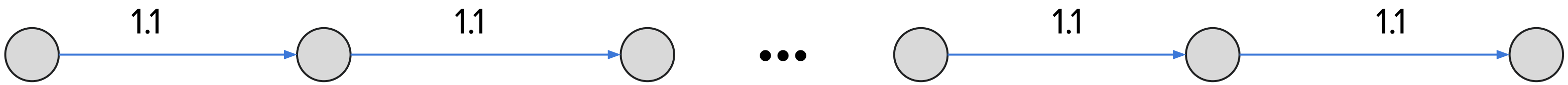


$$y = (1.1)^{100} x = 13780.6x$$

Initialisation matters

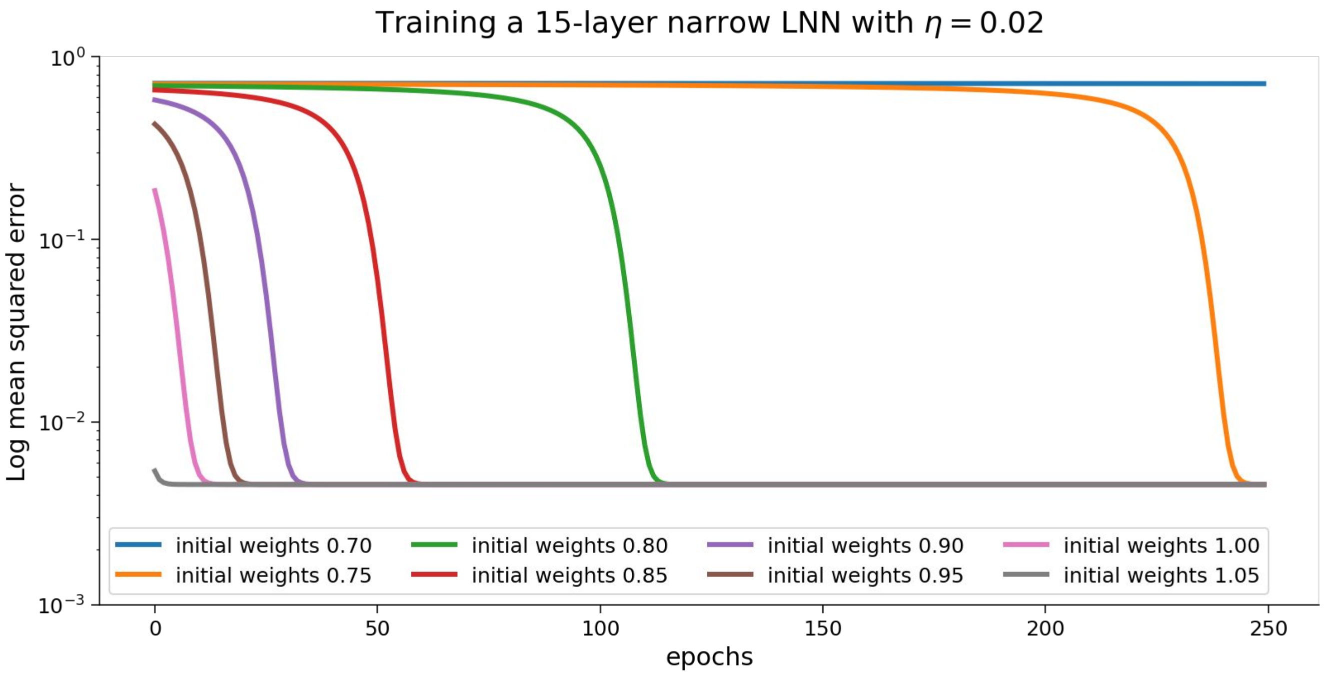
Unlike in shallow networks, learning in deep networks is exquisitely sensitive to initialisation

Basic reason: products of numbers vanish or explode $y = (\prod_{i=1}^{D} w_i)x$



Explore how initialisation impacts learning in a deep network.

Initialisation matters



Initialisation matters

Initialisations in deep networks need to be carefully chosen so that activity and gradients have similar magnitude across the network

Initialisations that preserve variance across depth are known as “dynamic isometry” initialisations

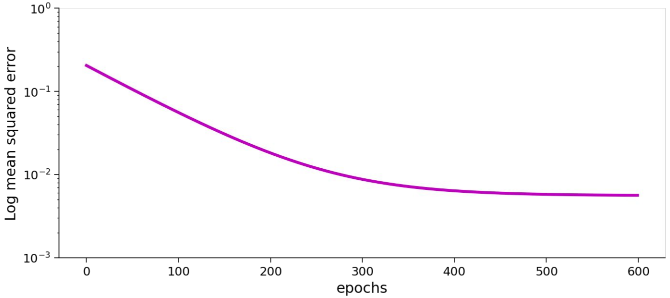
For deep narrow linear network, this corresponds to weights near 1

$$y = 1^{100} x = x$$

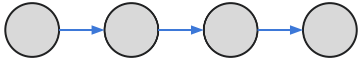
Wrap up: the effect of depth

Shallow

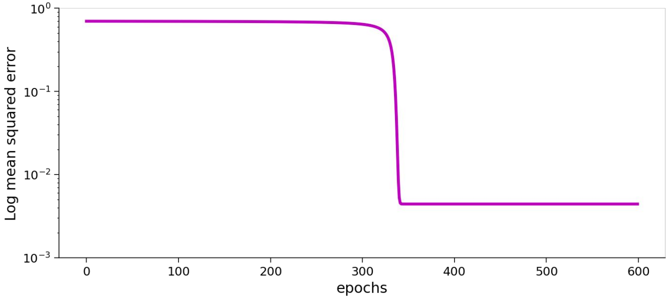
Training a 0-layer LNN with $\eta = 0.02$ initialized with $w_i = 0.9$



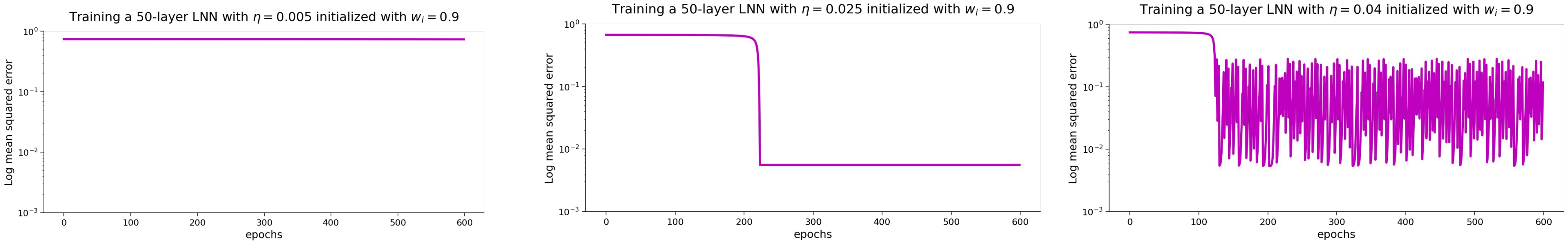
Deep



Training a 45-layer LNN with $\eta = 0.01$ initialized with $w_i = 0.9$



Wrap up: learning rate



Wrap up: initialization