

Machine Learning Principles

Optimization

Instructor: Tugce Gurbuz

Nov 4th, 2022

What is optimization?

What do we optimize?

How do we optimize?

What is optimization?

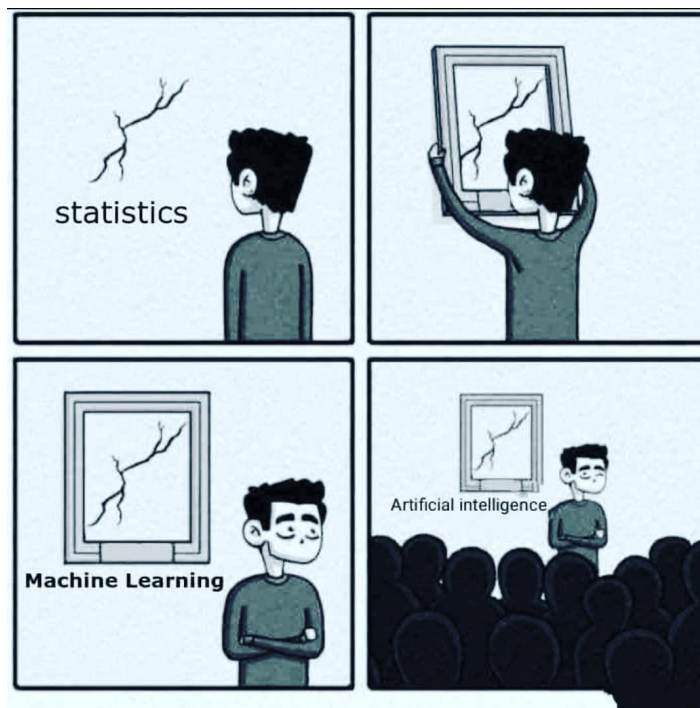
What do we optimize? -> **Parameters of the model to make the loss minimum**

How do we optimize?

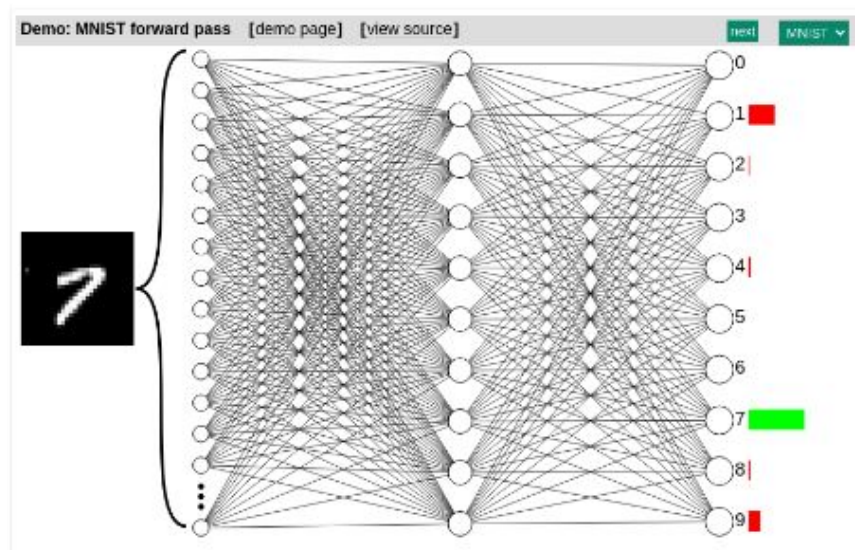
What is optimization?

What do we optimize? -> **Parameters of the model to make the loss minimum**

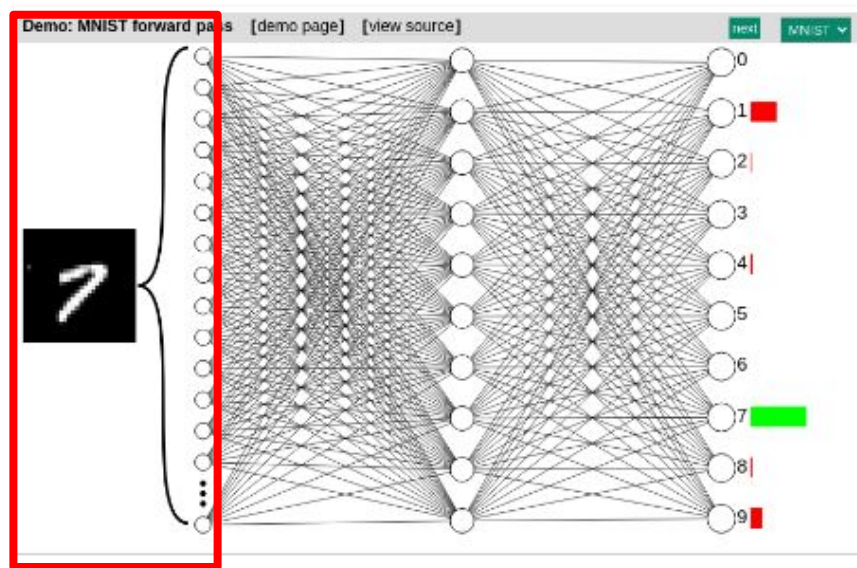
How do we optimize? -> **Math!**



Task in tutorial: Classifying handwritten digits with MLP

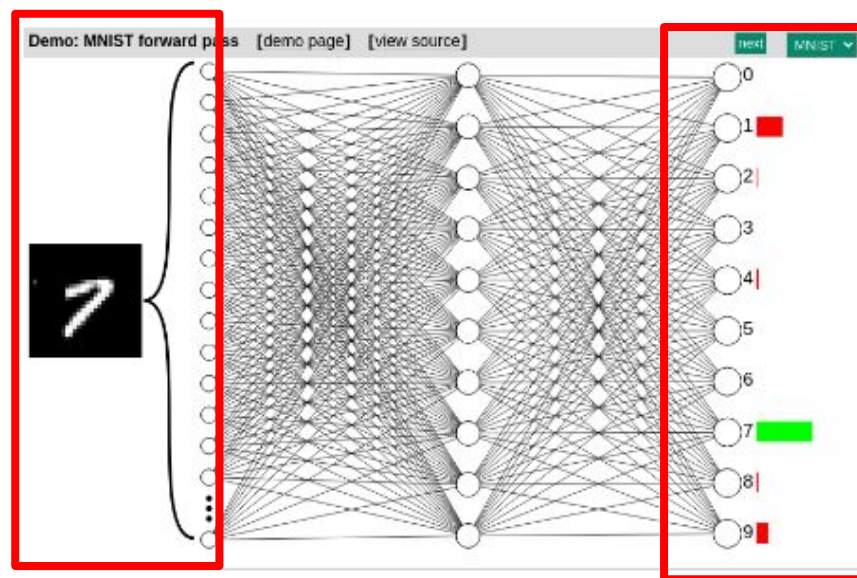


Task in tutorial: Classifying handwritten digits with MLP



28 x 28 image -> 784 vector

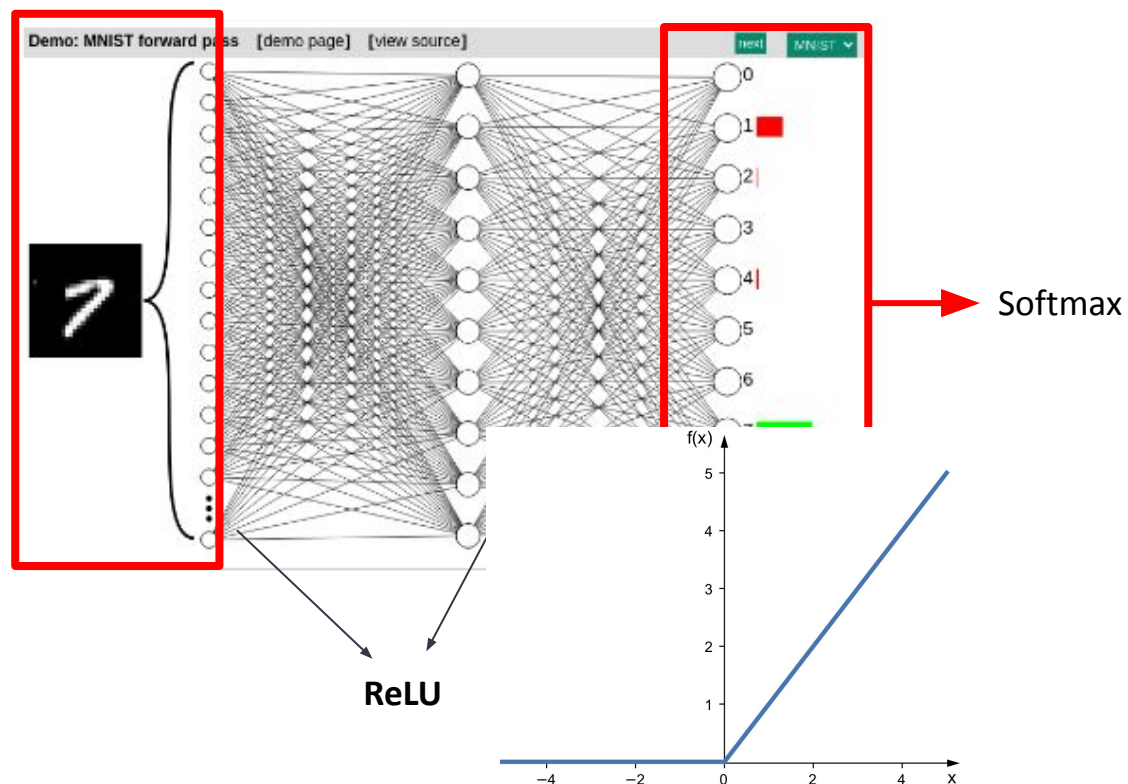
Task in tutorial: Classifying handwritten digits with MLP



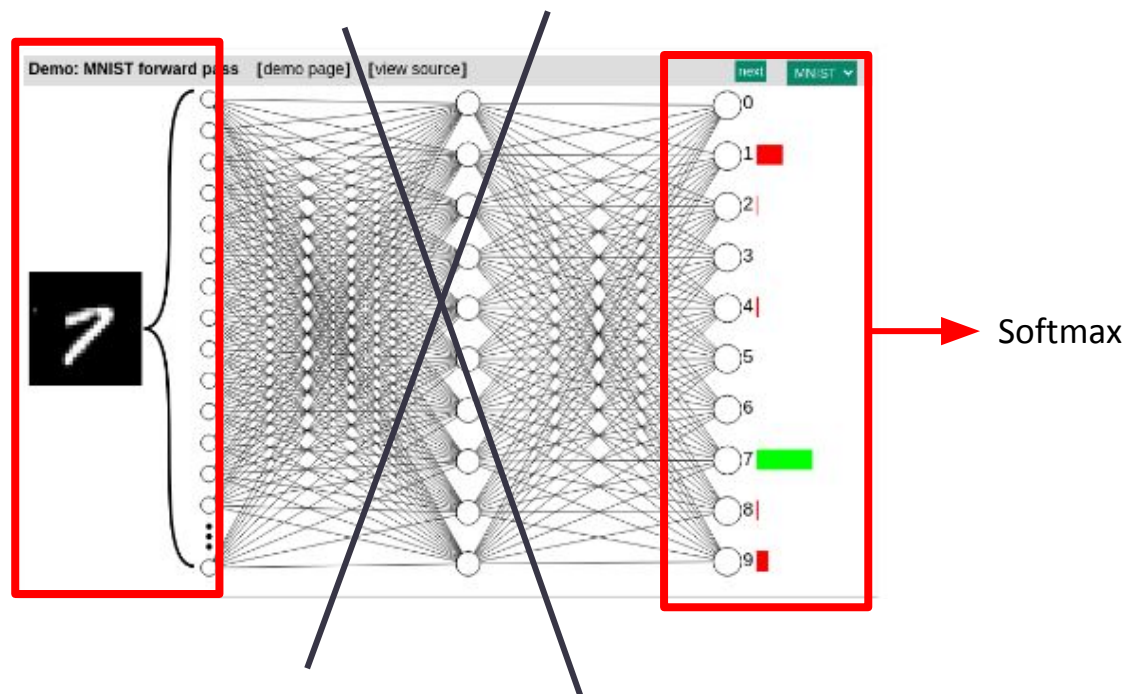
Softmax

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

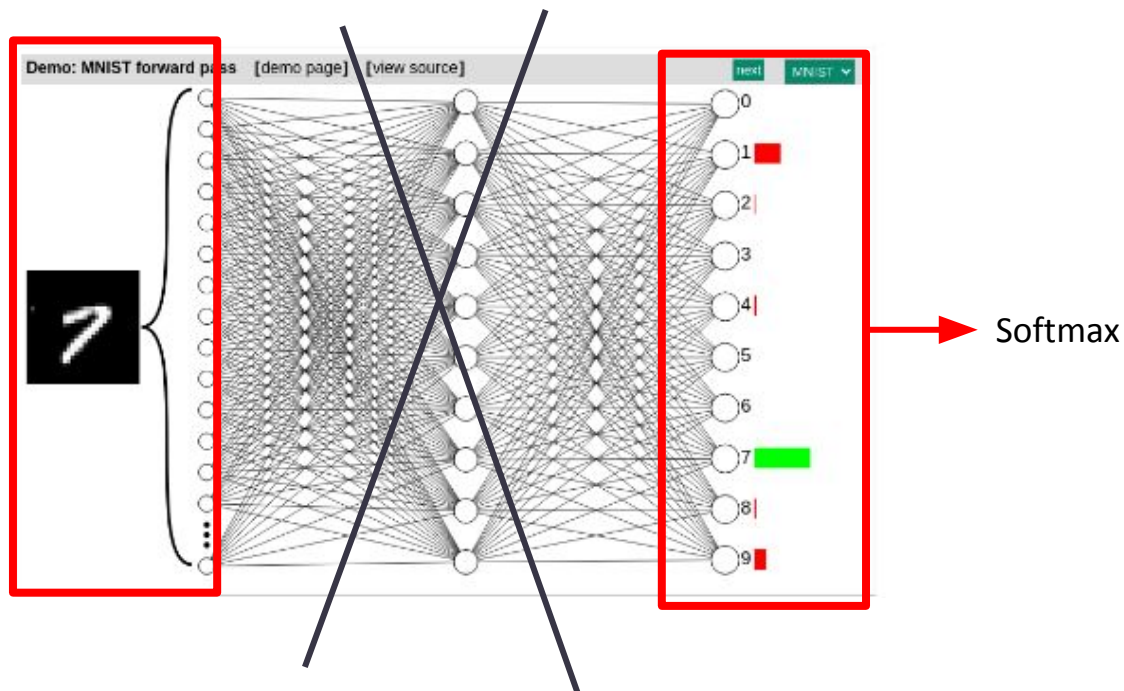
Task in tutorial: Classifying handwritten digits with MLP



Task in tutorial: Classifying handwritten digits with MLP



Task in tutorial: Classifying handwritten digits with MLP



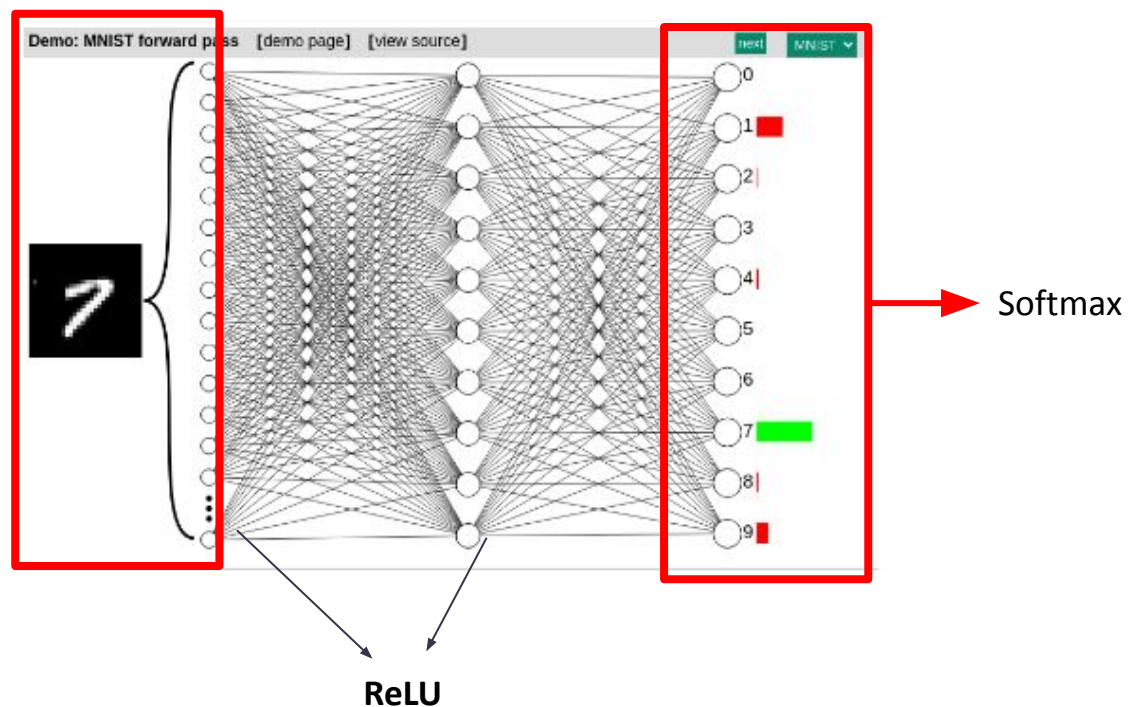
$$f(x) = \text{softmax}(Wx + b)$$

$$x \in \mathbb{R}^{784},$$

$$W \in \mathbb{R}^{10 \times 784}$$

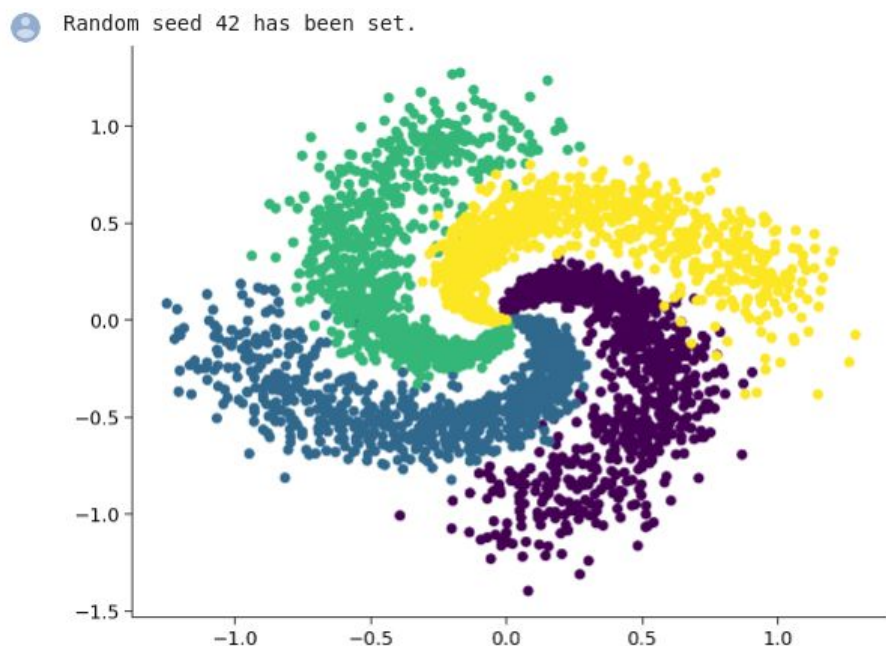
$$b \in \mathbb{R}^{10}$$

Task in tutorial: Classifying handwritten digits with MLP



Let's go to tutorial to implement our classification model!

How to calculate the loss?

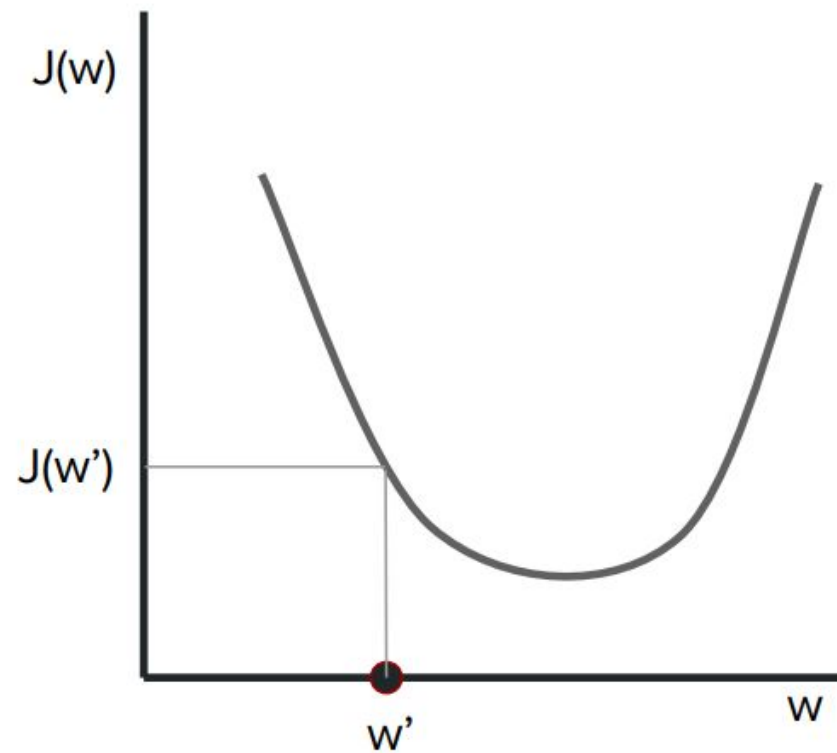


**Multiclass classification: Cross-entropy vs.
MSE?**

How to decide the loss function?

- Identify your problem (e.g., classification, regression?)
- Check the literature

How to minimize the loss?



How to minimize the loss?

Random search?

Algorithm:

- sample random points around current w

- if random point, w' , yields lower objective (i.e. $J(w') < J(w)$):

 - Accept w' as new position and store it in w

How to minimize the loss?

Random search?



How to minimize the loss?

Gradient Descent <3

Algorithm:

- Compute gradient (it points uphill)
- Do step in opposite direction of gradient
- Step size (learning rate), η



How to minimize the loss?

Gradient Descent

Algorithm:

- Compute gradient (it points uphill)
- Do step in opposite direction of gradient
- Step size (learning rate), η

$$w_{t+1} = w_t - \eta \nabla J(w_t)$$

How to minimize the loss?

Gradient Descent

Algorithm:

- Compute gradient (it points uphill)
- Do step in opposite direction of gradient
- Step size (learning rate), η

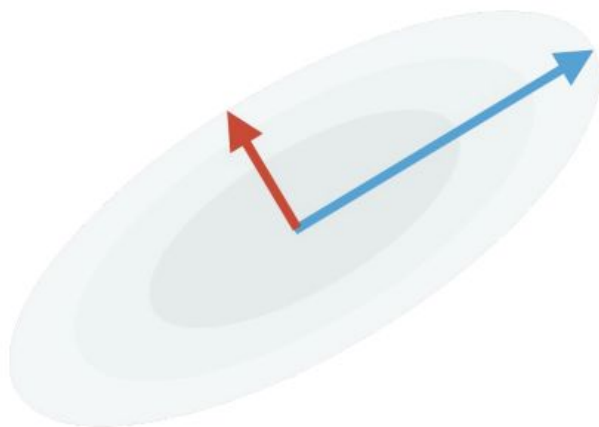
$$w_{t+1} = w_t - \eta \nabla J(w_t)$$

Let's go to our tutorial to implement Gradient Descent!!

How to minimize the loss?

Gradient Descent

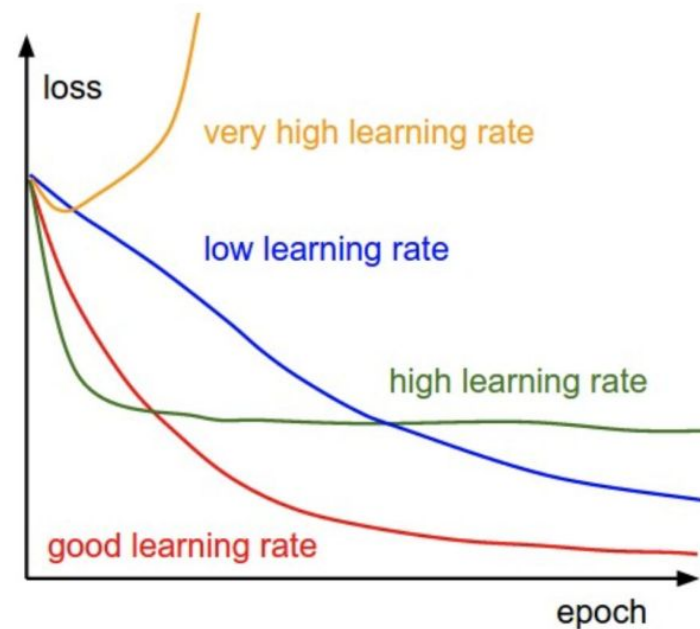
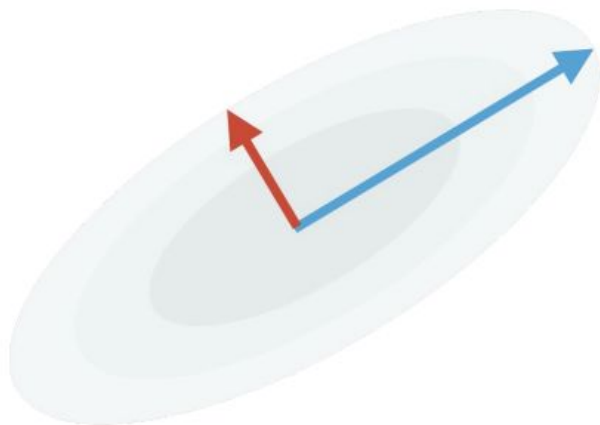
- How to choose learning rate?



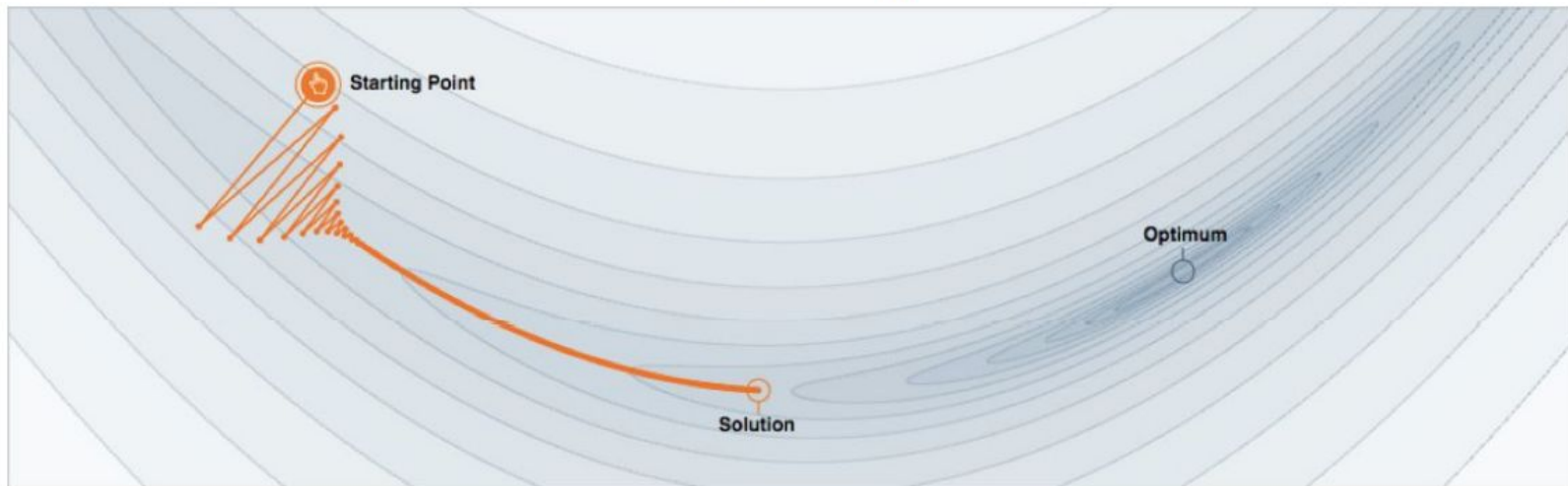
How to minimize the loss?

Gradient Descent

- How to choose learning rate?



How to minimize the loss?

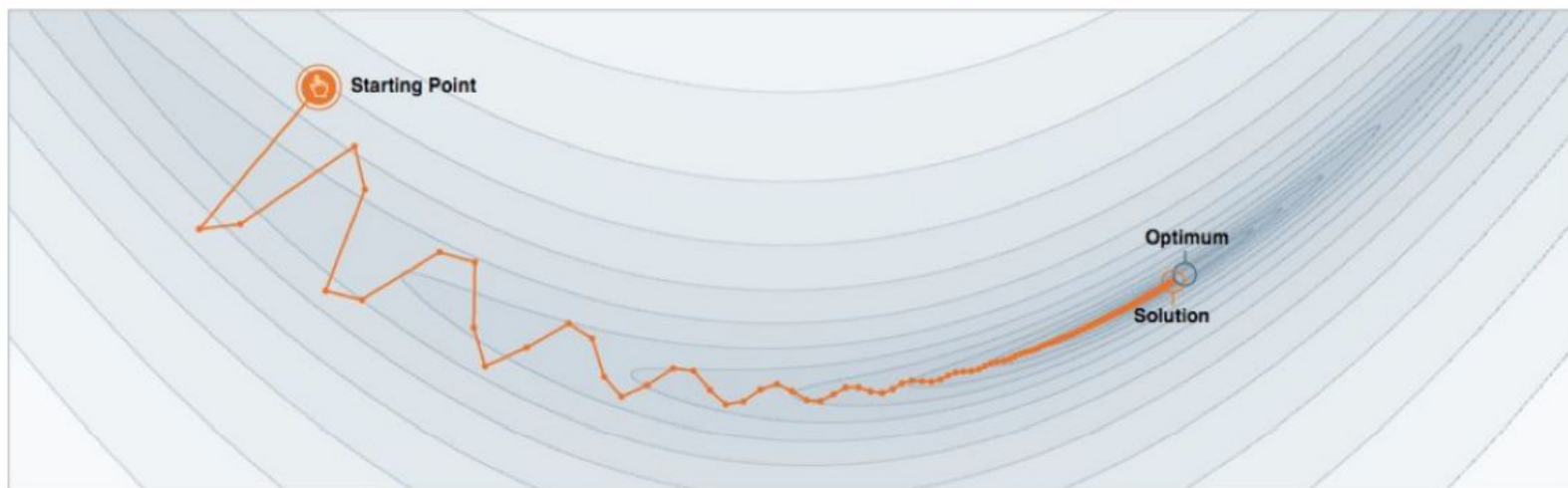


[Distill.pub]

How to minimize the loss?

Momentum < 3

- Accelerates along flat directions
- Slows down along sharp directions



[Distill.pub]

How to minimize the loss?

How does the momentum work?

Momentum algorithm:

- Do a gradient descent step
- Apply the update from the last iteration, only smaller (momentum step)

$$w_{t+1} = w_t - \eta \nabla J(w_t) + \beta (w_t - w_{t-1})$$

How to minimize the loss?

How does the momentum work?

Momentum algorithm:

- Do a gradient descent step
- Apply the update from the last iteration, only smaller (momentum step)

$$w_{t+1} = w_t - \eta \nabla J(w_t) + \beta (w_t - w_{t-1})$$

Let's go to our tutorial to implement Momentum!

How to minimize the loss?

Adaptive Methods <3

- Learning rate schedules

$$w_{t+1} = w_t - \eta_t \nabla J(w_t)$$

How to minimize the loss?

Adaptive Methods <3

- Learning rate schedules

$$w_{t+1} = w_t - \eta_t \nabla J(w_t)$$

Polynomial schedules, e.g. $\rightarrow \eta_t = \frac{\alpha}{c+t}, \eta_t = \frac{\alpha}{c+\sqrt{t}}$

Exponential

Stepwise decay

Cosine/cyclical schedules

How to minimize the loss?

Adaptive Methods <3

- **Adagrad (Duchi et al., 2011)**
 - Adapts the learning rate for each parameter
 - Using running sum squares of past gradients
 - Typically used in stochastic form:
 - Instead of full gradient, use gradient from mini-batch

$$[w_{t+1}]_i = [w_t]_i - \frac{\eta}{\sqrt{[v_{t+1}]_i + \epsilon}} [\nabla J(w_t)]_i$$

$$[v_{t+1}]_i = \sum_{s=1}^t [\nabla J(w_s)]_i^2$$

How to minimize the loss?

Adaptive Methods <3

- **RMSprop**
 - Uses a moving average instead of sum used by Adagrad
 - Moving average can be useful on non-convex objectives

$$[w_{t+1}]_i = [w_t]_i - \frac{\eta}{\sqrt{[v_{t+1}]_i + \epsilon}} [\nabla J(w_t)]_i$$

$$[v_{t+1}]_i = \alpha [v_t]_i + (1 - \alpha) [\nabla J(w_t)]_i^2$$

How to minimize the loss?

Adaptive Methods <3

- **RMSprop**
 - Uses a moving average instead of sum used by Adagrad
 - Moving average can be useful on non-convex objectives

$$[w_{t+1}]_i = [w_t]_i - \frac{\eta}{\sqrt{[v_{t+1}]_i + \epsilon}} [\nabla J(w_t)]_i$$

$$[v_{t+1}]_i = \alpha [v_t]_i + (1 - \alpha) [\nabla J(w_t)]_i^2$$

- **Adam: RMSprop + momentum**

How to minimize the loss?

Adaptive Methods <3

- **RMSprop**
 - Uses a moving average instead of sum used by Adagrad
 - Moving average can be useful on non-convex objectives

$$[w_{t+1}]_i = [w_t]_i - \frac{\eta}{\sqrt{[v_{t+1}]_i + \epsilon}} [\nabla J(w_t)]_i$$

$$[v_{t+1}]_i = \alpha [v_t]_i + (1 - \alpha) [\nabla J(w_t)]_i^2$$

- **Adam: RMSprop + momentum**

Let's go to our tutorial to implement RMSprop!

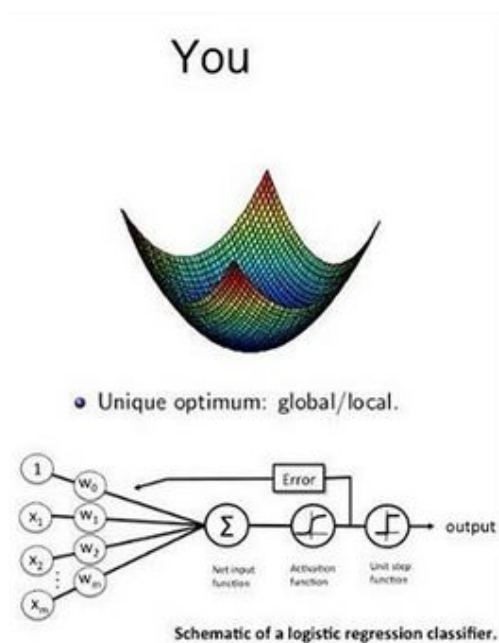
How to minimize the loss?

Non-convexity Problem

How to minimize the loss?

Non-convexity Problem

- Convex < 3 -> have the same global and local minimum
- Non-convex -> have different global and local minimum

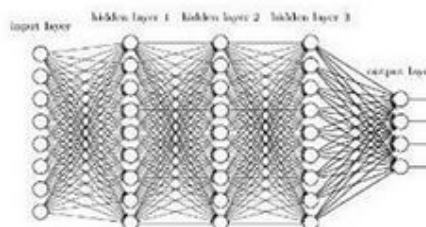


The guy she tells you
not to worry about



- Multiple local optima
- In high dimensions possibly

Deep neural network



How to minimize the loss?

Non-convexity Problem

- Convex \rightarrow have the same global and local minimum
- Non-convex \rightarrow have different global and local minimum

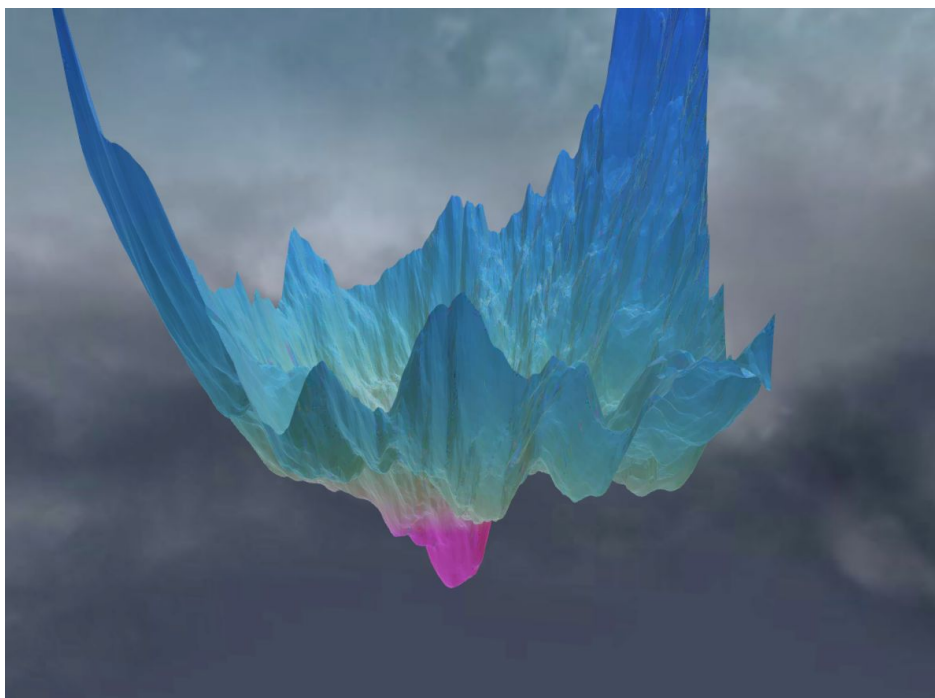
Great non-convexity, comes with great responsibility!

How to minimize the loss?

Non-convexity Problem

- Initialization matters!!

<https://losslandscape.com/explorer>



How to minimize the loss?

Non-convexity Problem

- Overparameterization to rescue!

Let's go to tutorial and overparameterize our MLP!

How to minimize the loss?

Computation Cost Problem

How to minimize the loss?

Computation Cost Problem

- Minibatch training < 3

How to minimize the loss?

Computation Cost Problem

- Minibatch training \rightarrow stochastic gradient descent

How to minimize the loss?

Computation Cost Problem

- Minibatch training \rightarrow stochastic gradient descent
 - Minibatch size:
 - Too small batch size: optimization bounces around a lot, and can lead to slower convergence to a minimum.
 - Too big batch size: won't fit on GPU
 - Rule of thumb \rightarrow pick the largest batch size that fits in the GPU

How to minimize the loss?

Computation Cost Problem

- Minibatch training \rightarrow stochastic gradient descent
 - Minibatch size:
 - Too small batch size: optimization bounces around a lot, and can lead to slower convergence to a minimum.
 - Too big batch size: won't fit on GPU
 - Rule of thumb \rightarrow pick the largest batch size that fits in the GPU

Let's go to our tutorial and implement Mini Batch Training!!!