A gentle Introduction to deep learning with JAX

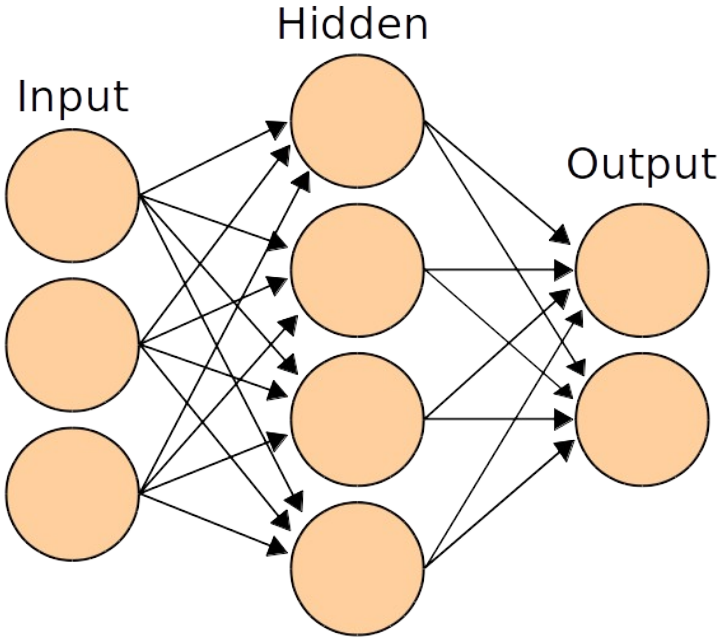
BAMB! Summer School
Tutorial 5

We fill focus on **regression models**
From inputs to outputs

function parameters

$$y = f(x, \theta)$$

output input

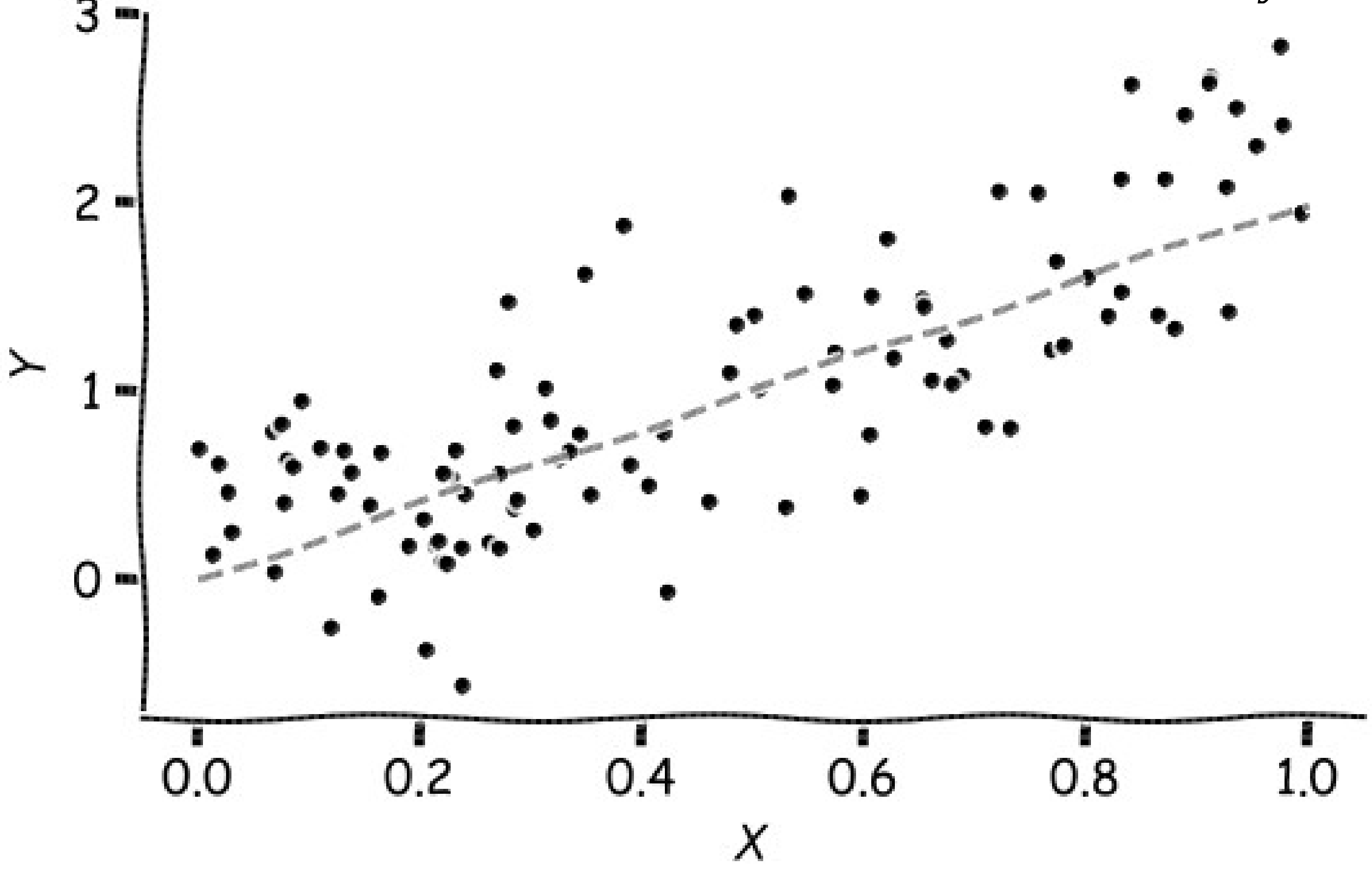


Learning by minimizing a loss

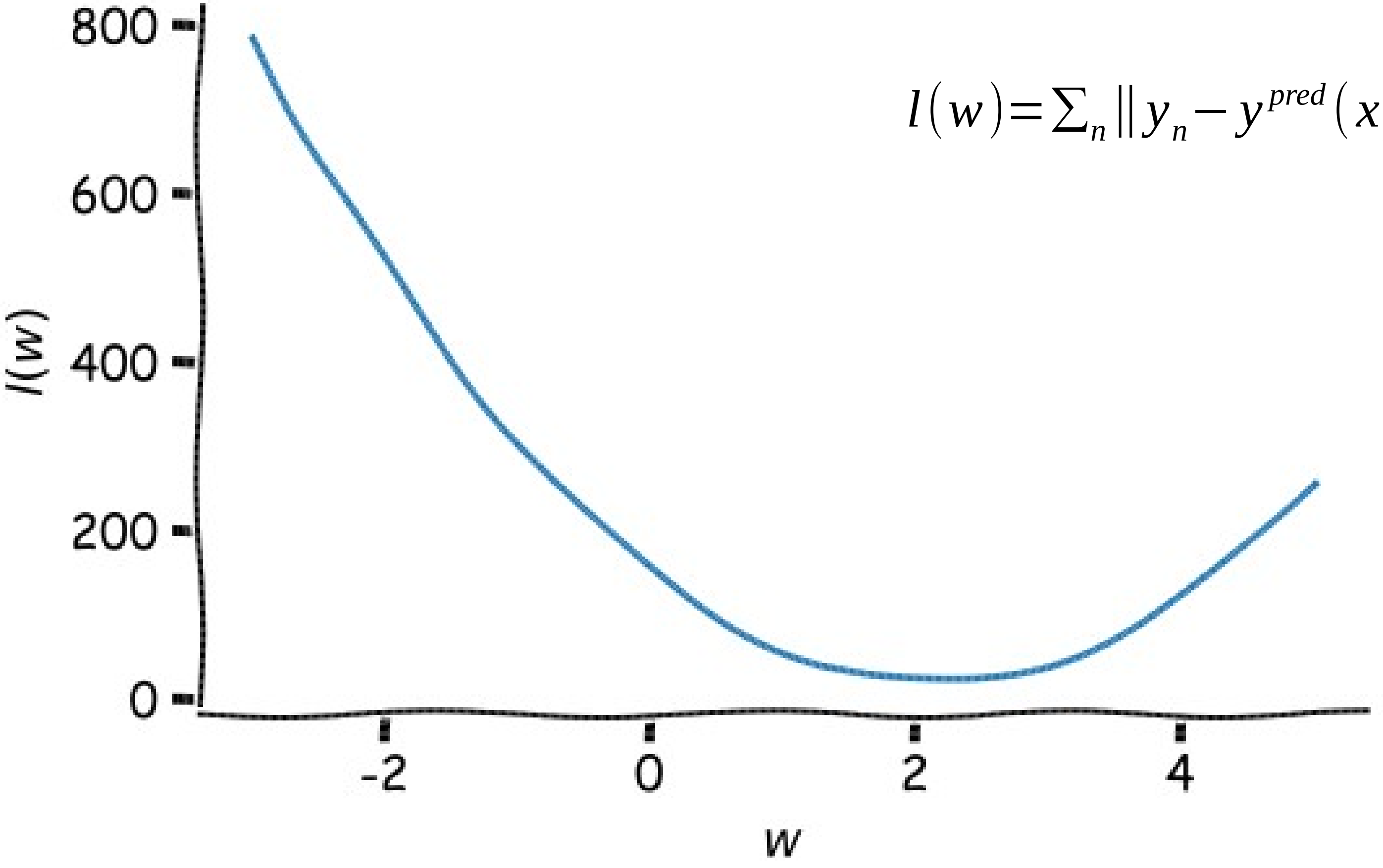
$$l(\theta)=\sum_n \|y_n - f(x,\theta)\|^2$$

Gradient Based Optimization

$y = w\,x + b + noise$

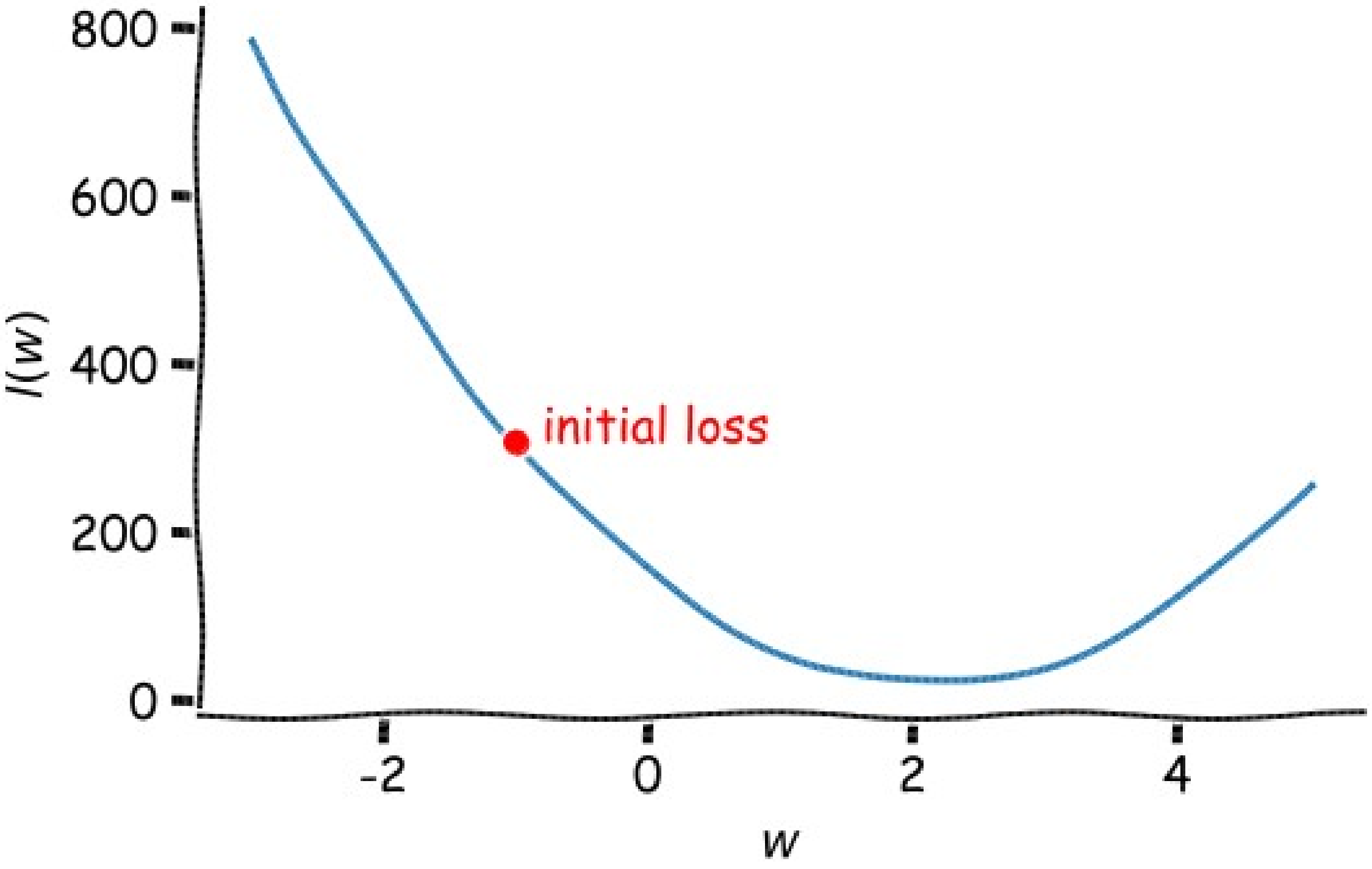


Gradient Based Optimization

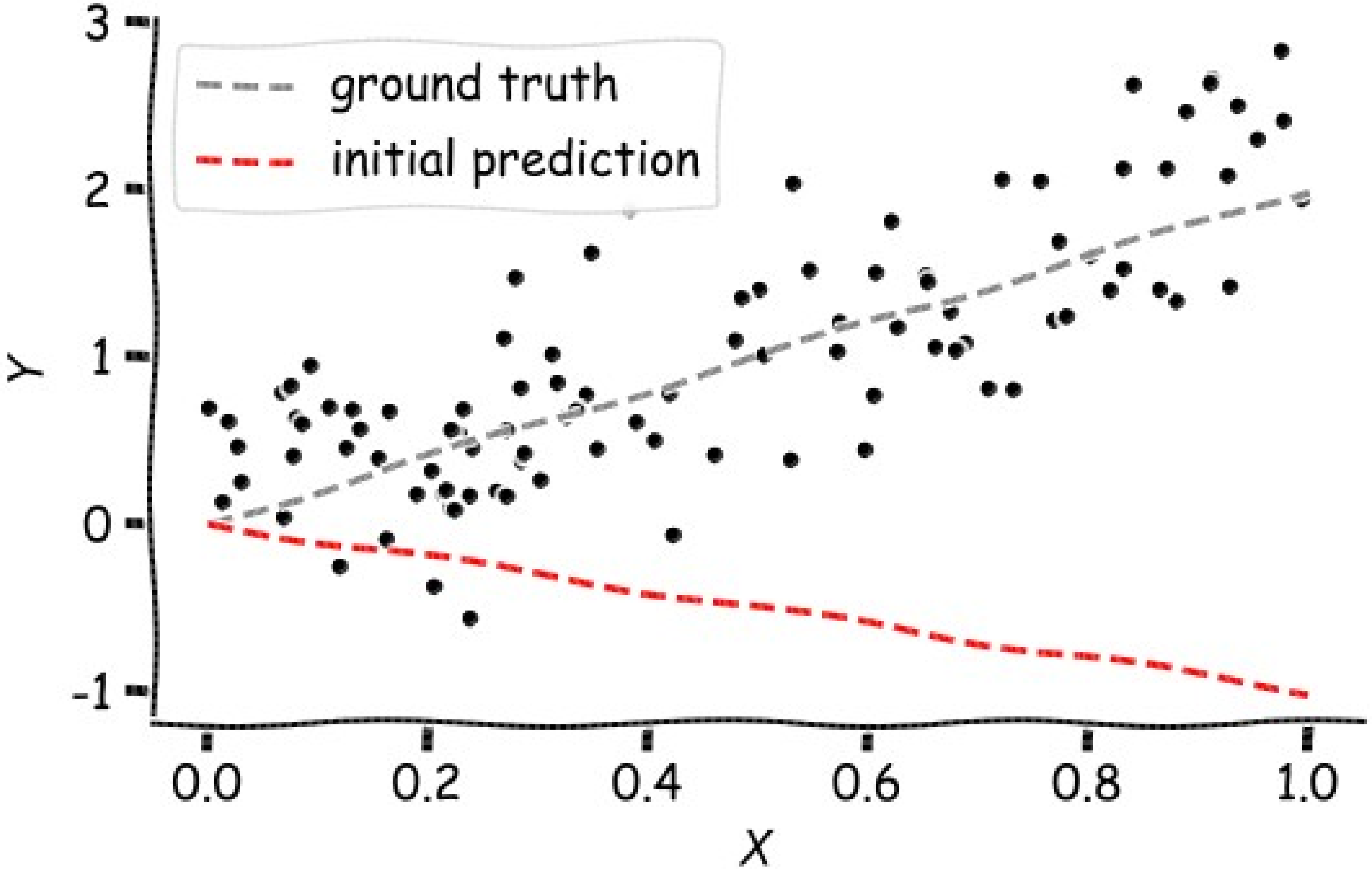


$$l(w) = \sum_n \| y_n - y^{pred}(x_n) \|^2$$

Gradient Based Optimization

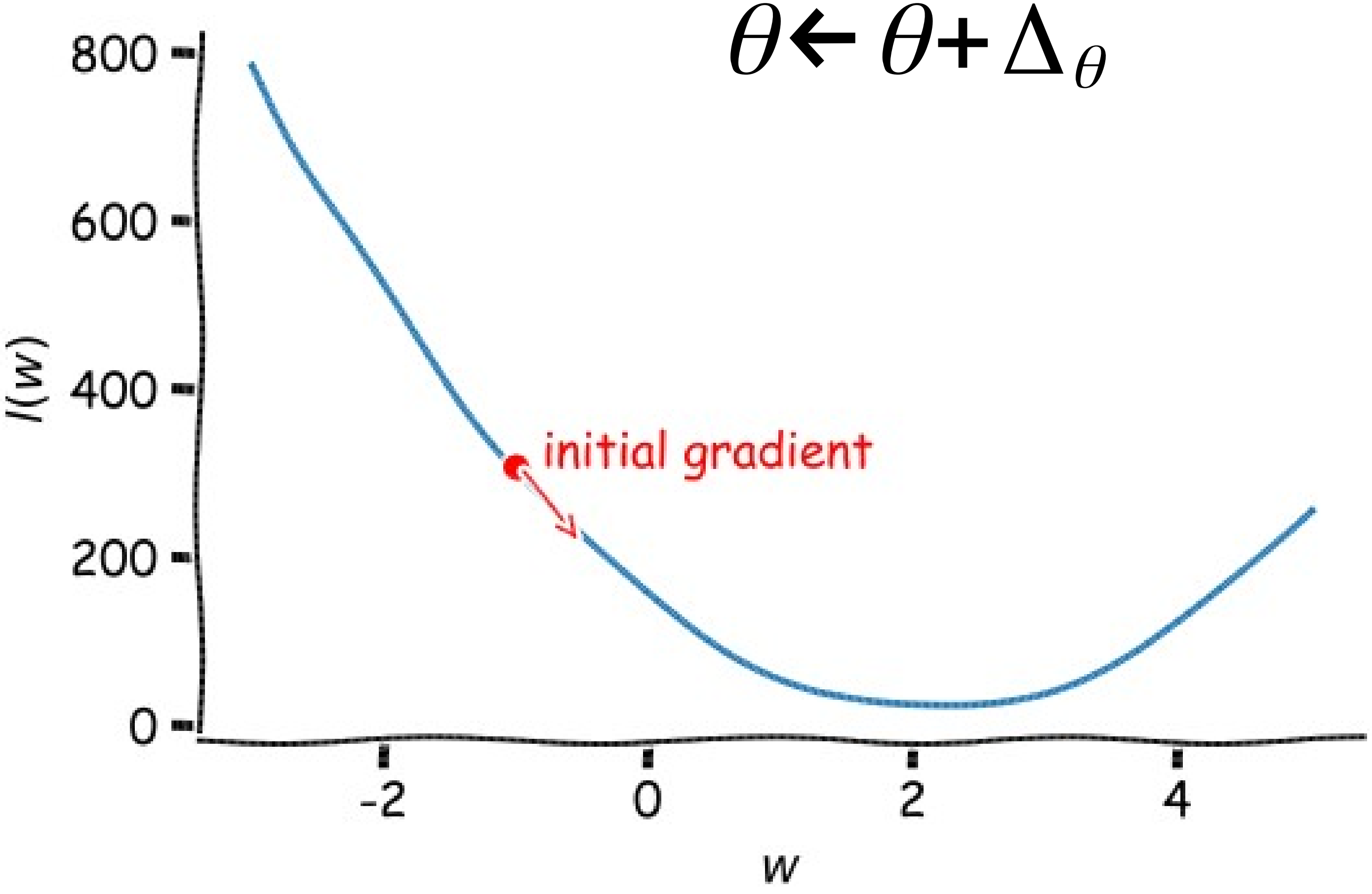


Gradient Based Optimization

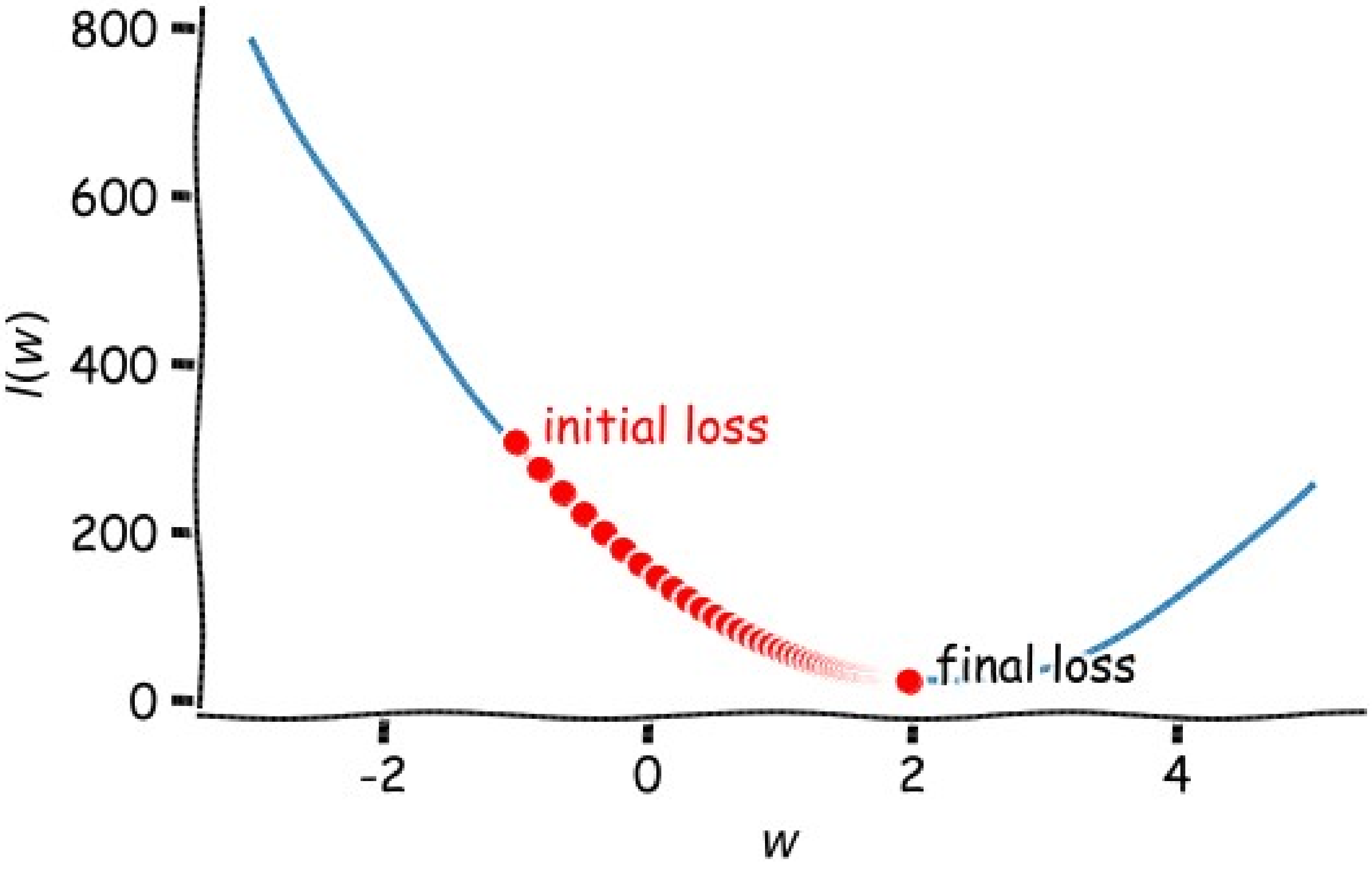


Gradient Based Optimization

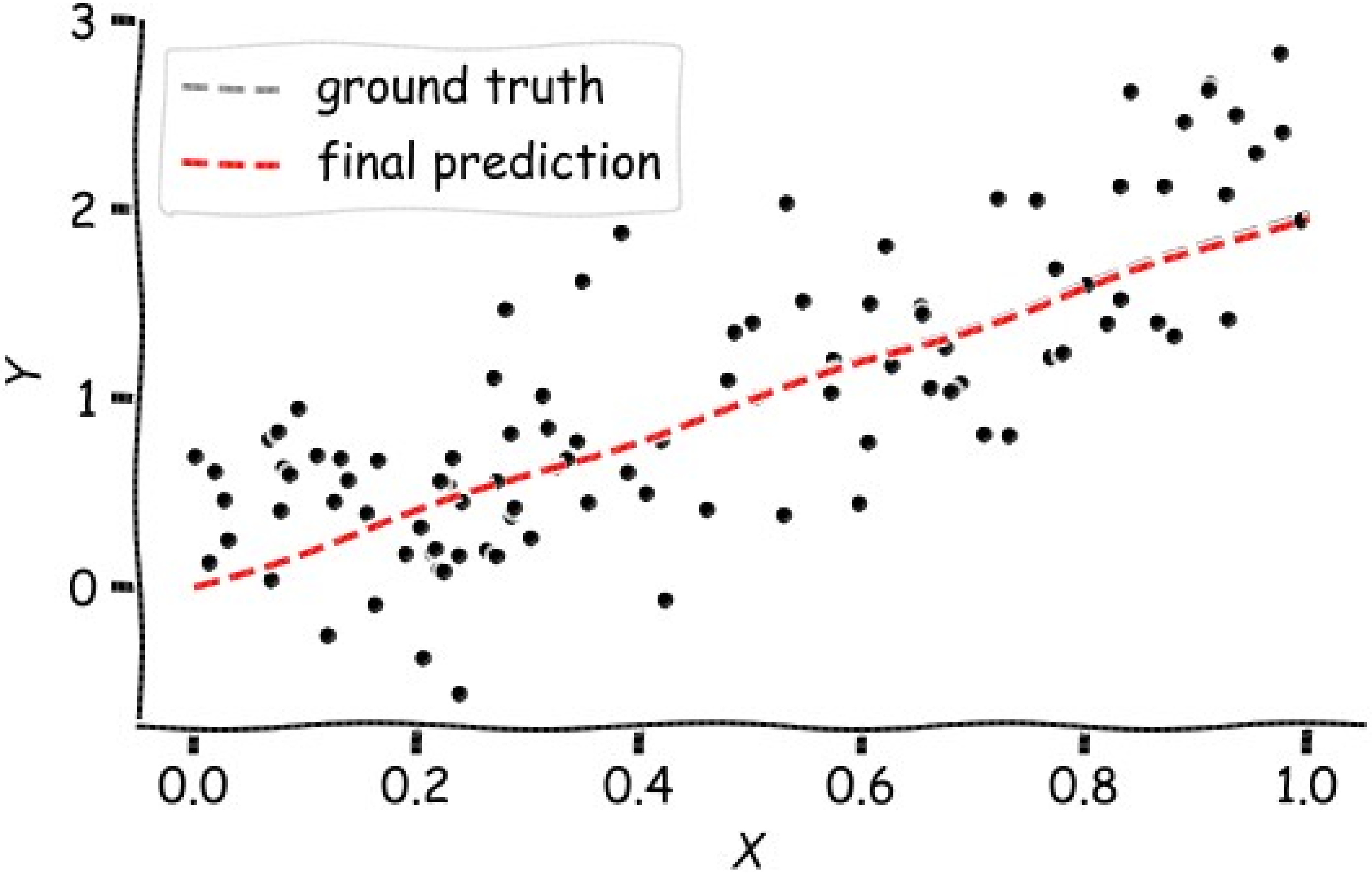
$$\theta \leftarrow \theta + \Delta_{\theta}$$



Gradient Based Optimization

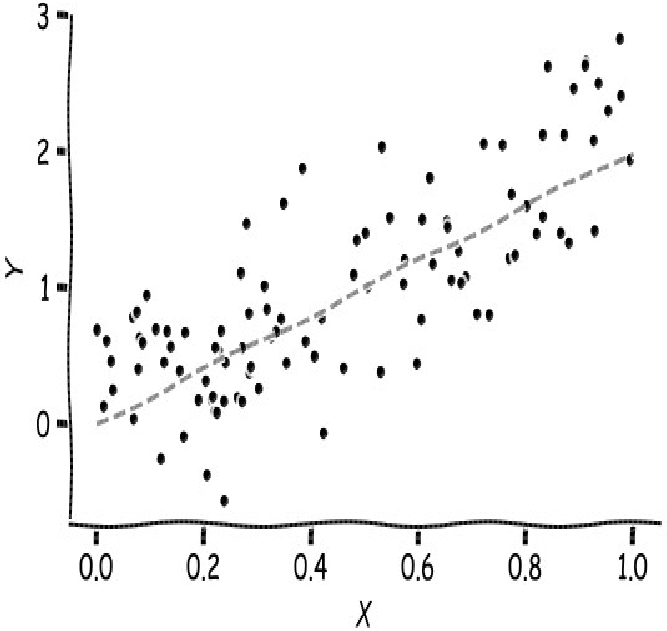


Gradient Based Optimization



We will work with different models

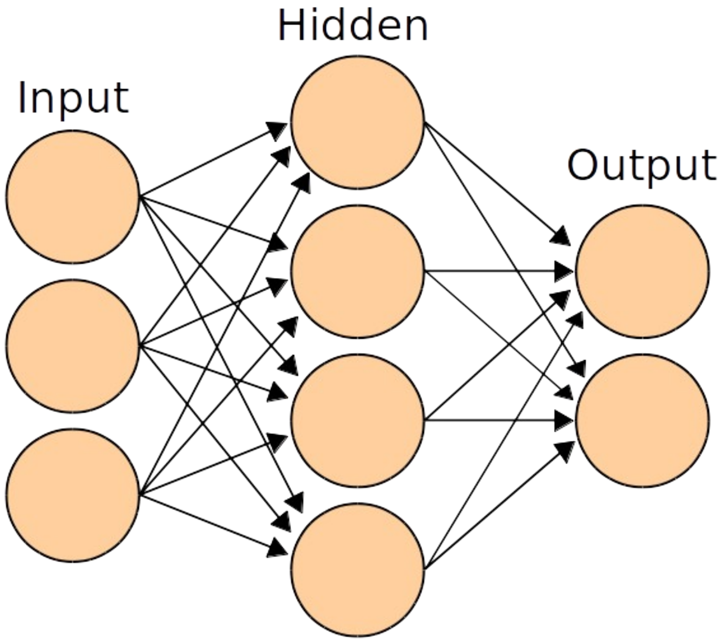
- Linear / Logistic Regression (as a warm up)
- Neural networks (NN) for image classification
- Convolutional neural networks (CNN)



Let’s get started coding!

What is JAX for?

- Perform **Gradient Based** optimization …
- … in LARGE models (millions of parameters)



A bit about JAX

JAX is a library

***** in Python

***** … made to be very close to the python library **numpy**

***** that computes gradient automatically for you (**automatic differentiation**)

***** that can compile code on demand to be very fast (**just in time** [or jit] compilation)

***** that has useful features to ‘vectorize’ function : **vmap**
(apply a function designed for scalars to vectors or tensors)

Some resources for the less experienced coders

A course on using python for scientific computing
http://ucl-cs-grad.github.io/scipython/

Numpy for matlab (by Numpy creators)
https://numpy.org/doc/stable/user/numpy-for-matlab-users.html

Numpy for matlab users cheat sheet:
https://mas-dse.github.io/DSE200/cheat_sheets/1_python/6_2_NumPy_for_MATLAB_users.pdf

Tutorial on github

https://github.com/vincentadam87/intro_to_jax