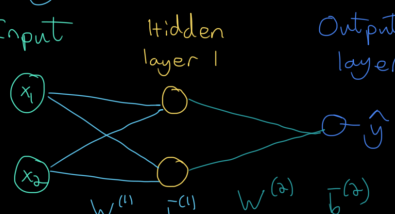
Training neural networks

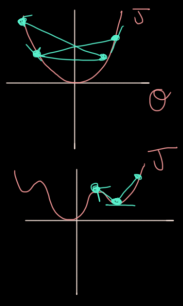


Training regression model

Choose objective function $J = MSE = \frac{1}{N}\sum_i (y_i - \hat{y}_i)^2$

Initialize weights and biases

Train using extension of gradient descent



$\frac{\partial J}{\partial W^{(1)}}, \frac{\partial J}{\partial b^{(1)}}, \frac{\partial J}{\partial W^{(2)}}, \frac{\partial J}{\partial b^{(2)}}$ } backpropagation (later)

$W^{(1)}_{it+1} = W^{(1)}_{it} - \alpha \frac{\partial J}{\partial W^{(1)}}$ } computed on all training data

↑ learning rate

⋮

Stochastic gradient descent

Use some (not all) of training data at each step

minibatch

Input: 10K images Output: 10K neural responses

training data

At each iteration compute gradient on minibatch of 200 images and update parameters

$\frac{10K}{200} = 50$

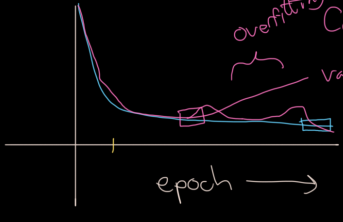
For each pass of the data: ← epoch

For each minibatch (50 in all): ← shuffle ordering

→ Compute gradients

→ Update parameters

→ Record loss function



Compute loss on validation data

How do we know when to stop?

→ pick # of epochs ahead of time

→ when training loss barely decreases

might overfit unless we regularize well

→ when validation loss stops decreasing

Early stopping regularization