Deep Learning

For supervised learning on non-linear models

Terminology

dataset $\{(x^{(i)}, y^{(i)})\}_{i=1}^{n}$ $\quad x^{(i)} \in \mathbb{R}^d, \; y^{(i)} \in \mathbb{R}$

$h_\theta(x): \mathbb{R}^d \to \mathbb{R}$

Cost / loss fn:

$$J^{(i)}(\theta) = (y^{(i)} - h_\theta(x))^2$$ mean-squared loss

$$J(\theta) = \frac{1}{n} \sum_{i=1}^{n} J^{(i)}(\theta)$$

Optimization objective

$$\min_\theta J(\theta)$$

$\Rightarrow$ use gradient descent

$$\theta := \theta - \alpha \nabla J(\theta)$$

Stochastic Gradient Descent

for $i = 1 \dots n_{iter}$

sample j from $\{1 \dots n\}$ uniformly

$$\theta := \theta - \alpha \nabla J^{(i)}(\theta)$$

Minibatch SGD

$\rightarrow$ Computing B gradients $\nabla^{(j_1)}(\theta) \dots \nabla^{(j_B)}(\theta)$ can be faster than individual computation

for $i = 1 \dots n_{iter}$

sample B examples $\{j_1 \dots j_B\}$ from $\{1 \dots n\}$ uniformly w/o replacement

$$\theta := \theta - \frac{\alpha}{B} \sum_{k=1}^{B} \nabla J^{(j_k)}(\theta)$$

Note: SGD only finds local optimum

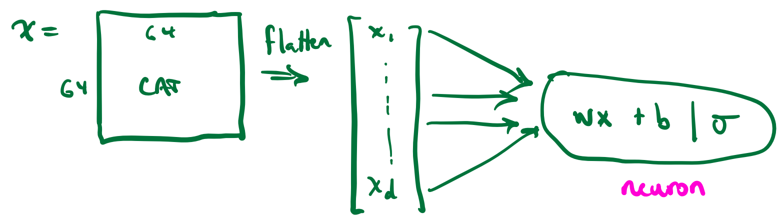
BUT: research has shown that the quality of these optima are very good compared to global

Building up a neural network

Starting from logistic regression

e.g. Goal: find cats in images

1 → presence of cat
0 → absence of class



$$\hat{y} = \sigma(\theta^T x) = \sigma(\underbrace{w}_{1\times 12{,}288}\underbrace{x}_{12{,}288\times 1} + b)$$

$d = 64 \times 64 \times 3 = 12{,}288$

i) Initialize $\underbrace{w}_{\text{weights}}$, $\underbrace{b}_{\text{bias}}$

ii) Find optimal w, b

iii) Use $\hat{y} = \sigma(wx+b)$ to predict

$$\mathcal{L} = -\left[y \log \hat{y} + (1-y)\log(1-\hat{y})\right]$$

$$w := w - \alpha \frac{\partial \mathcal{L}}{\partial w}$$

$$\sigma' = \sigma(1-\sigma)$$

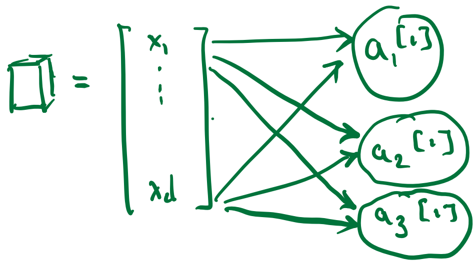
$$b := b - \alpha \frac{\partial \mathcal{L}}{\partial b}$$

$$\#\text{parameters} = |w| + |b| = 12{,}288 + 1$$

define neuron — linear + activation

model — architecture (where neurons are, how they are arranged) + parameters

Goal 2.0: Find cat/lion/iguana in images



$$\hat{y}_1 = a_1^{[1]} = \sigma(w_1^{[1]} + b_1^{[1]})$$

$$\hat{y}_2 = a_2^{[1]} = \sigma(w_2^{[1]} + b_2^{[1]})$$

$$\hat{y}_3 = a_3^{[1]} = \sigma(w_3^{[1]} + b_3^{[1]})$$

Notation: $[1] \equiv$ layer

subscripts identify neuron within layer

$a_2^{[1]}$ ← 1st layer, ← 2nd neuron in layer

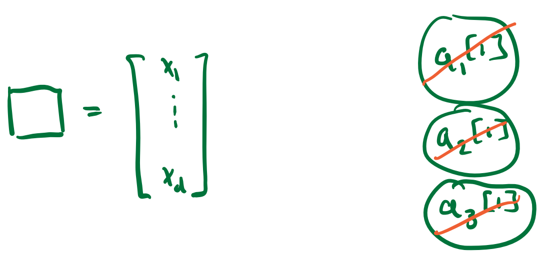
params: $3(d+1)$

Dataset: images + labels

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{array}{l} \text{cat} \\ \text{lion} \\ \text{iguana} \end{array}$$

one-hot encoding

Goal 3.0: add constraint: unique animal in image



$z_1^{[1]}$ $\quad e^{z_1^{[1]}} / \sum_{k=1}^{3} z_k^{[1]}$

$z_2^{[1]}$ $\quad e^{z_2^{[1]}} / \sum_{k=1}^{3} z_k^{[1]}$

$z_3^{[1]}$ $\quad e^{z_3^{[1]}} / \sum_{k=1}^{3} z_k^{[1]}$

$$\hat{y}_1 = \sigma(\underbrace{w_1^{[1]}x + b_1^{[1]}}_{z_1^{[1]}})$$

#params: $3(d+1)$

Training

$$\mathcal{L}_{3N} = -\sum_{k=1}^{3} \left[y_k \log \hat{y}_k + (1-y_k)\log(1-\hat{y}_k) \right]$$

$$\mathcal{L}_{CE} = -\sum_{k=1}^{3} y_k \log \hat{y}_k$$

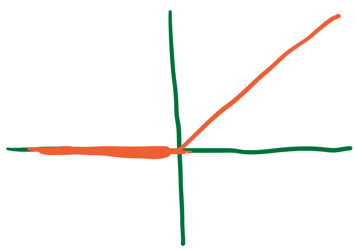
cross-entropy

Predicting age of the cat

Options:

1. Bucket ages, several neurons to predict
2. Change activation fn

linear fn: $f(x) = x$ linear regression



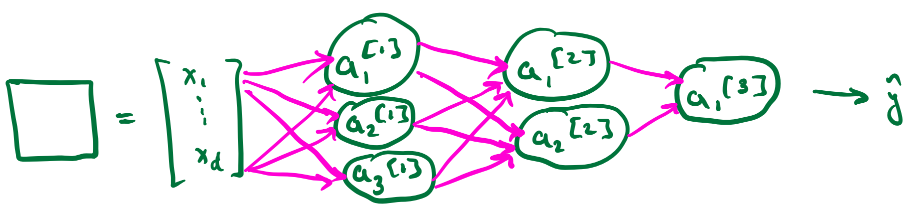
Rectified Linear Unit ReLU

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$$

Modified loss fn $\|\hat{y} - y\|_1$, $\|\hat{y} - y\|_2^2$

Neural Networks, more generally

goal: image $\Rightarrow$ cat or no cat

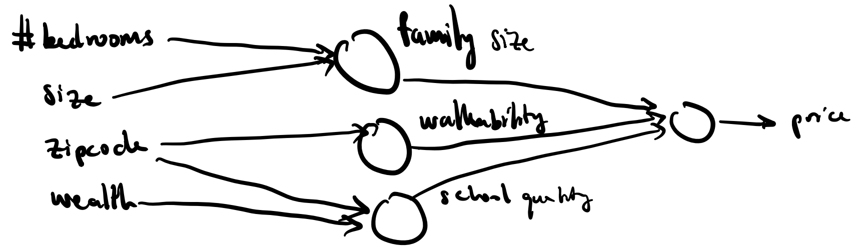


params 3d+3 2·3+2 1·2+1

input layer hidden layer output layer

House price prediction

Data:



Propagation equations

$$z^{[1]} = \overset{3\times d}{W^{[1]}}\, \overset{d\times 1}{x} + \overset{3\times 1}{b^{[1]}} \qquad 3\times 1$$

$$Z^{[1]} = W^{[1]} X + b^{[1]}$$
$$\overset{3\times n}{Z^{[1]}} = \overset{3\times d}{W^{[1]}}\, \overset{d\times n}{X} + \overset{3\times 1}{b^{[1]}}$$

Broadcasting: $\tilde{b}^{[1]} = [\underbrace{b^{[1]} \dots b^{[1]}}_{n}]$

$$a^{[1]} = \sigma(z^{[1]}) \qquad 3\times 1$$

$$z^{[2]} = \underset{2\times 3}{W^{[2]}}\, \underset{3\times 1}{a^{[1]}} + \underset{2\times 1}{b^{[2]}} \qquad 2\times 1$$

$$a^{[2]} = \sigma(z^{[2]}) \qquad 2\times 1$$

$$z^{[3]} = \underset{1\times 2}{W^{[3]}}\, \underset{2\times 1}{a^{[2]}} + \underset{1\times 1}{b^{[3]}} \qquad 1\times 1$$

$$a^{[3]} = \sigma(z^{[3]}) \qquad 1\times 1$$

$$X = \left[x^{(1)} \dots x^{(n)} \right]$$

[ ] layer
( ) id of example
capital: batch of examples

Optimize $W^{[1]}$ $W^{[2]}$ $W^{[3]}$ $b^{[1]}$ $b^{[2]}$ $b^{[3]}$