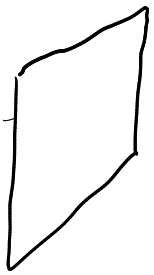
MAP: maximize w/ respect to $\vec{w}$

$$P(\vec{w} \mid D) = \frac{P(D \mid \vec{w})\, P(\vec{w})}{P(D)}$$

$\vec{w}$: model parameters

GOAL: find "best" set of params $\vec{w}^*$

example: machine learning classification



input: image (100 x 100 px x 3 colors)

$\Rightarrow$ input vector $\vec{x}$ (length 3×10^4) $\rightarrow$ function (neural network) $\rightarrow$ prob. vector $\vec{p} = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}$

p_i = prob. that input is category i

2 categories (i.e. cat or dog)

Neural network: dense neural net (DNN)

or multilayer perceptron

Series of operations:

"one layer" of net

$$g\left(A^{(1)} \vec{x} + \vec{b}^{(1)}\right) = \vec{z}^{(1)}$$

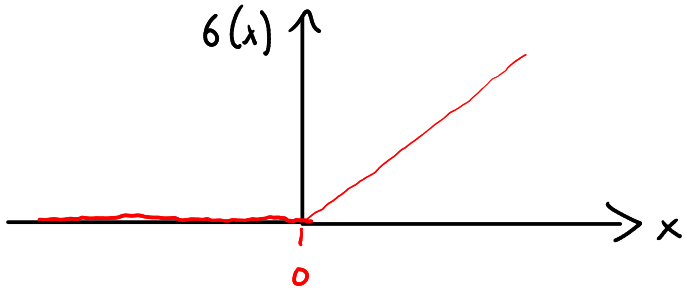
$A^{(1)}$: matrix (could be rectangular)

$\vec{b}^{(1)}$: vector

$\sigma(x)$ = nonlinear "threshold" function

example: ReLU $\sigma(x) = \max(0, x)$

$\sigma(\vec{x})$
$= (\sigma(x_1), \sigma(x_2), \ldots)$



next layer: $\vec{z}^{(2)} = \sigma\left(A^{(2)} \vec{z}^{(1)} + \vec{b}^{(2)}\right)$

$\vec{z}^{(n)} = \sigma\left(A^{(n)} \vec{z}^{(n-1)} + \vec{b}^{(n)}\right)$

↓

same dim as # categories (here: 2)

overall process depends on components: $A^{(i)}, \vec{b}^{(i)}$ $i = 1, \ldots, n$

$\Rightarrow$ vector of model params $\vec{w}$

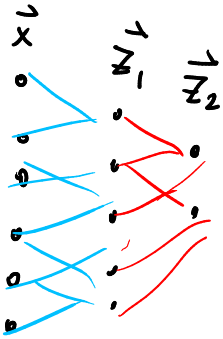
final step: "softmax"

convert $\vec{z}^{(n)}$ into prob. $\vec{p}$

$$p_i = \frac{e^{-z_i^{(n)}}}{e^{-z_1^{(n)}} + e^{-z_2^{(n)}} + \ldots}$$

$0 \le p_i \le 1$

$\sum_i p_i = 1$

$\vec{x} \Rightarrow$  $\Rightarrow \vec{p}(\vec{x}; \vec{w})$

soft max

universal approx. theorems:

for DNNs if # layers & size of layers is large enough

$\Rightarrow$ you can approx. any func. that takes input from a finite domain

---

Bayesian interpretation:

dataset $\mathcal{D}$ "training data" for net

"image"	label:
$\vec{x}^{(1)}$	$\ell_1 = \begin{cases} 1 & \text{cat} \\ 2 & \text{dog} \end{cases}$
$\vdots$	
$\vec{x}^{(N)}$	ℓ_N

$$\mathcal{D} = \{ (\vec{x}^{(i)}, \ell_i) \mid i = 1, \ldots, N \}$$

interested in $\mathcal{P}(\vec{w} \mid \mathcal{D})$

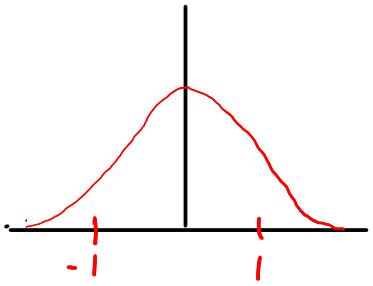
first step: calculate $\mathcal{P}(\mathcal{D} \mid \vec{w})$

$$\mathcal{P}(\mathcal{D} \mid \vec{w}) = \prod_{i=1}^{N} p_{\ell_i}(\vec{x}^{(i)}; \vec{w}) \leq 1$$

$= 1$ when you have a perfect classifier

Want to maximize $\mathcal{P}(\vec{w} \mid \mathcal{D}) = \dfrac{\mathcal{P}(\mathcal{D} \mid \vec{w})\,\mathcal{P}(\vec{w})}{\mathcal{P}(\mathcal{D})}$ ← prior

"regularization": assume net params are not too big in magnitude (to prevent blow-ups)



$$\mathcal{P}(w_i) = \frac{1}{\sqrt{2\pi}} e^{-w_i^2/2}$$

Gaussian of width 1

$$\mathcal{P}(\vec{w}) = \prod_j \mathcal{P}(w_j)$$

trick: to maximize $\mathcal{P}(\vec{w} \mid \mathcal{D})$ we take -log of it & minimize:

loss func.

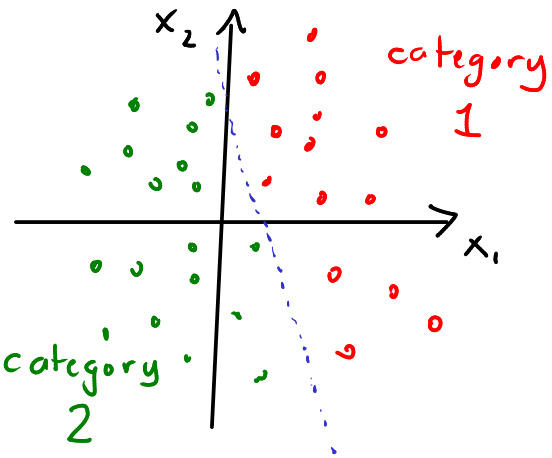
$$\mathcal{L}(\vec{w}) = -\log \mathcal{P}(\vec{w} \mid \mathcal{D})$$

$$= -\log \mathcal{P}(\mathcal{D} \mid \vec{w}) - \log \mathcal{P}(\vec{w}) + \underbrace{\log \mathcal{P}(\mathcal{D})}_{\text{constant (ignore)}}$$

↓

we can minimize w/ gradient descent

prob. 2 of PS #1



$\vec{x} = (x_1, x_2)$

$\Downarrow$

$p_1(\vec{x}; \vec{w})$

= prob. that $\vec{x}$ is category 1

$$p_2(\vec{x}; \vec{w}) = 1 - p_1(\vec{x}; \vec{w})$$