TODAY

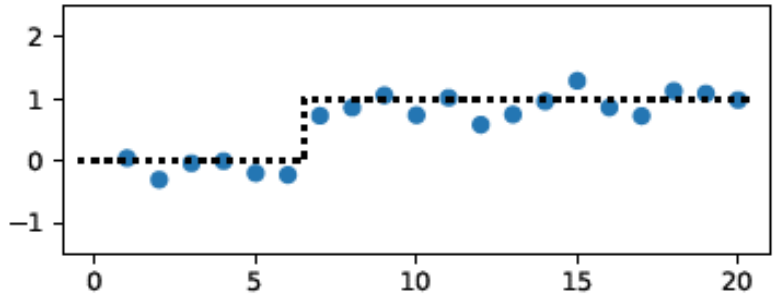
Mock exam question 1

Please collect a printout from the front bench on your left.

(a) A 0/1 signal is being transmitted. The transmitted signal at timeslot $i \in \{1, \ldots, n\}$ is $x_i \in \{0, 1\}$, and we have been told that this signal starts at 0 and then flips to 1, i.e. there is a parameter $\theta \in \{1, \ldots, n-1\}$ such that $x_i = 1_{i>\theta}$. The value of this parameter is unknown. The channel is noisy, and the received signal in timeslot i is

$$Y_i \sim x_i + \text{Normal}(0, \varepsilon^2)$$

where ε is known.



(i) Given received signals $(y_1, \ldots, y_n)$, find an expression for the log likelihood, $\log \Pr(y_1, \ldots, y_n \,;\, \theta)$. Explain your working. [5 marks]

(ii) Give pseudocode for finding the maximum likelihood estimator $\hat{\theta}$. [5 marks]

1. Skim read for keywords. What's the topic?
2. Look for question words. What is it asking you to do?
3. Think through the course. What sections are relevant?

(*b*) I have been monitoring average annual river levels for many years, and I have collected a dataset $(z_0, \ldots, z_n)$ where z_i is the level in year i since I started monitoring. I believe that for the first few years the level each year was roughly what it was the previous year, plus or minus some random variation; but that some year a drought started, and since then the level has decreased on average each year. I would like to estimate when the drought started. I do not know the other parameters.

(*i*) Propose a probability model for my dataset. [5 marks]

(*ii*) Explain how to fit your model. [5 marks]

This "propose a probability model" is open-ended and scary. How should we even begin to think about it?

1. Skim read for keywords. What's the topic?
2. Look for question words. What is it asking you to do?
3. Think through the course. What sections are relevant?
4. Read the whole question. What's the link?

Part (a) gave us a hammer. Can we see part (b) as a nail?

Deep learning*

* non-examinable

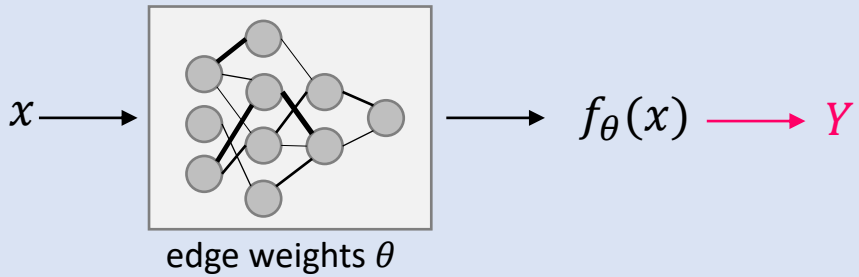
Supervised Learning

Data: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

Labels: $y_1, y_2, \ldots, y_n$

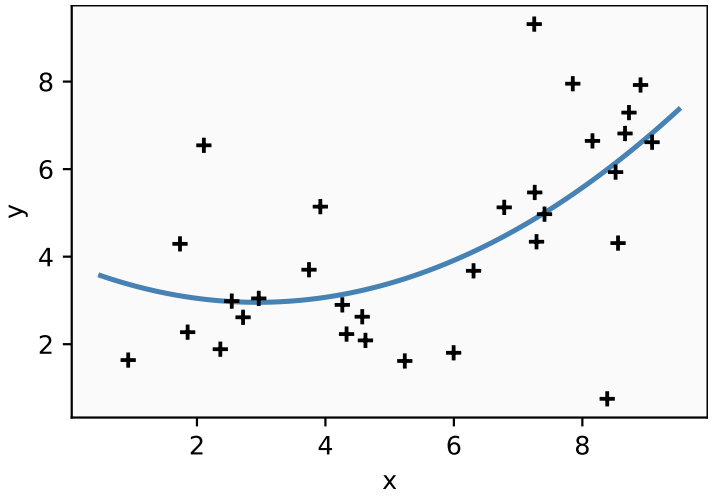
Task: fit the probability model $\Pr_Y(y\,;f_\theta(x))$

Training goal: MLE



Example (regression)

Given a labelled dataset consisting of pairs (x_i, y_i) of real numbers, fit the model $Y_i \sim \alpha + \beta x_i + \gamma x_i^2 + N(0, \sigma^2)$



Model for a single observation:

$$Y \sim \alpha + \beta x + \gamma x^2 + N(0, \sigma^2)$$
$$\sim N(\alpha + \beta x + \gamma x^2, \sigma^2)$$

Likelihood of a single observation:

$$\Pr_Y(y\,; x, \alpha, \beta, \gamma, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(y-(\alpha+\beta x+\gamma x^2))/2\sigma^2}$$

Log likelihood of the dataset:

$$\log \Pr(y_1, \ldots y_n;\ \alpha, \beta, \gamma, \sigma) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

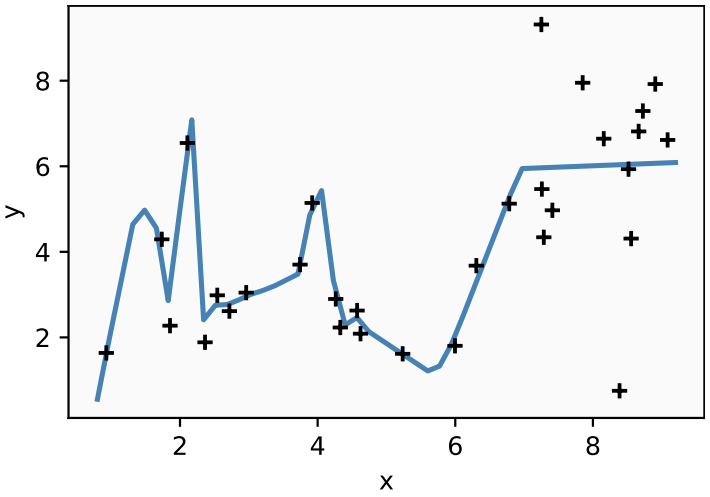
where $\hat{y}_i = \alpha + \beta x_i + \gamma x_i^2$

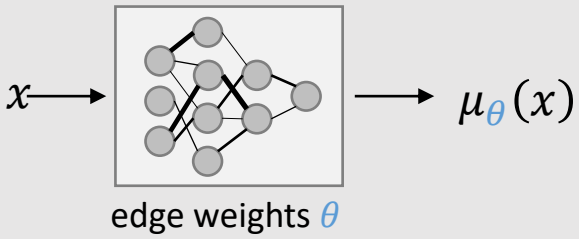
Optimize over the unknown parameters:

Example (regression)

Given a labelled dataset consisting of pairs (x_i, y_i) of real numbers, fit the model $Y_i \sim \mu_\theta(x_i) + N(0, \sigma^2)$.

(Here $\mu_\theta(\cdot)$ is some specified function with unknown parameters θ.)





edge weights θ

Log likelihood of the dataset:

$$\log \Pr(y_1, \ldots y_n;\ \theta, \sigma) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu_\theta(x_i))^2$$

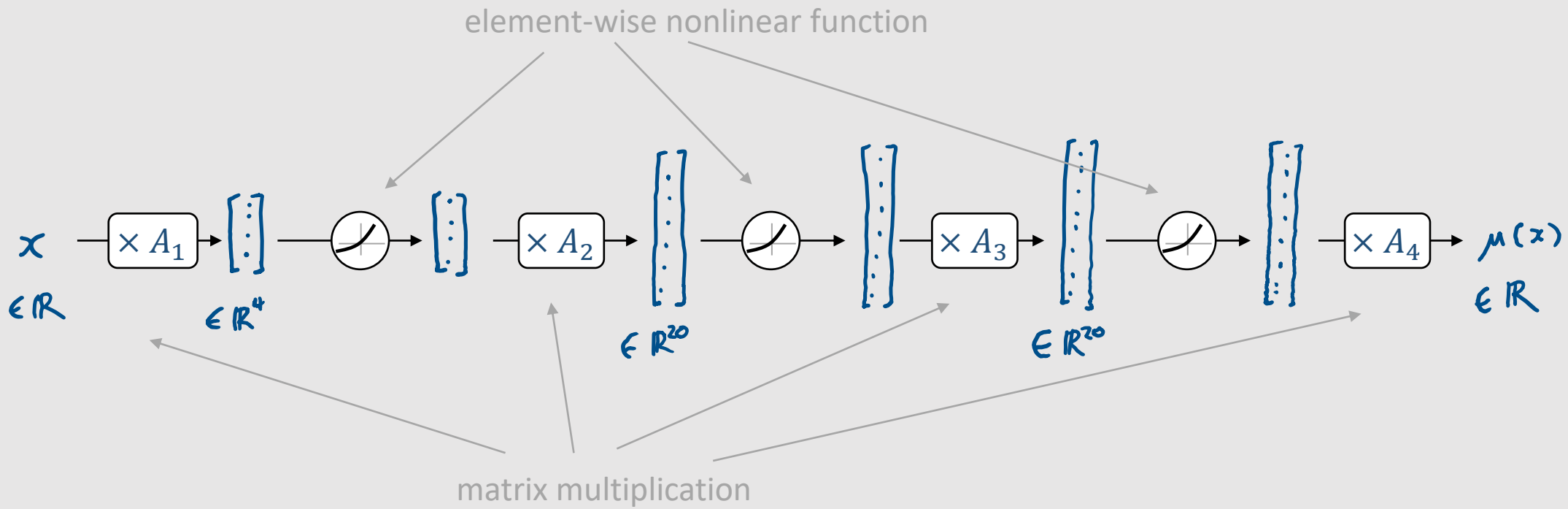
Optimize over the unknown parameters θ and σ:

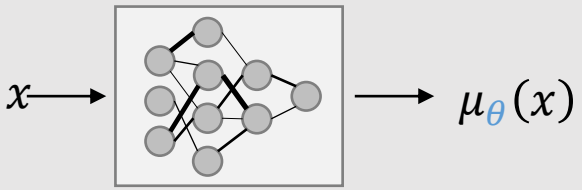
```
1  class RWiggle(nn.Module):
2      def __init__(self):
3          super().__init__()
4          self.μ = ...
5          self.σ = nn.Parameter(torch.tensor(1.0))
       # compute log Pr(y;x)
6      def forward(self, y, x):
7          σ2 = self.σ ** 2
8          return - 0.5*torch.log(2*π*σ2) - torch.pow(y - self.μ(x), 2) / (2*σ2)

9  x,y = ...
10 mymodel = RWiggle()

11 optimizer = optim.Adam(mymodel.parameters())
12 for epoch in range(10000):
13     optimizer.zero_grad()
14     loglik = torch.sum(mymodel(y, x))
15     (-loglik).backward()
16     optimizer.step()
```

See section 3.3 of printed notes. Or work through the tutorial [to be released tonight].

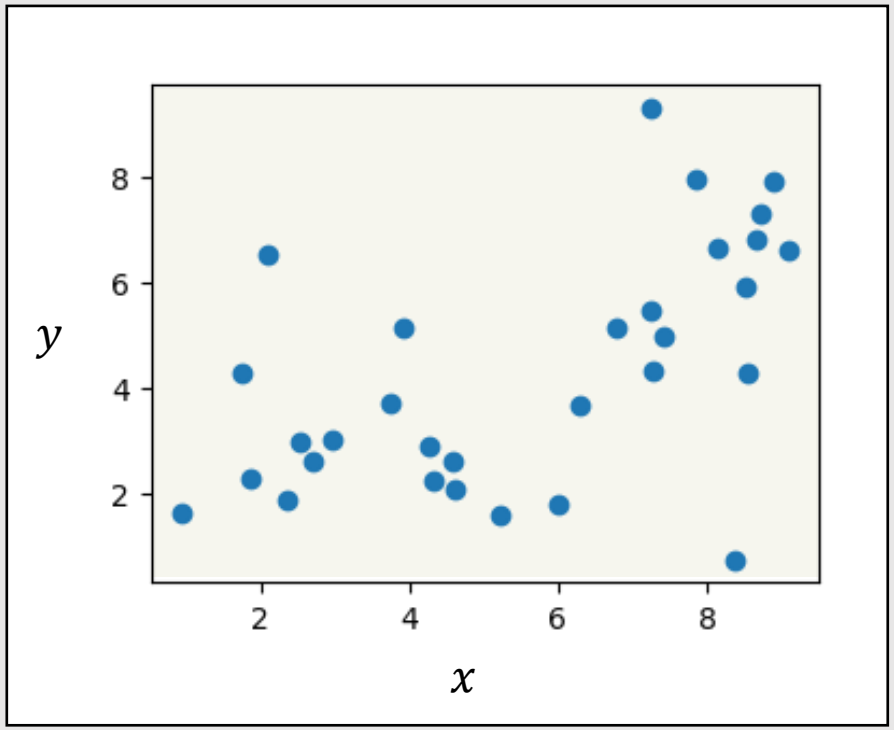





```
self.μ = nn.Sequential(
    nn.Linear(1,4), nn.LeakyReLU(),
    nn.Linear(4,20), nn.LeakyReLU(),
    nn.Linear(20,20), nn.LeakyReLU(),
    nn.Linear(20,1)  )
```

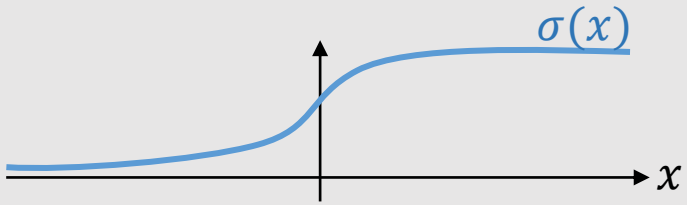
https://xkcd.com/1838

How do linear algebra & nonlinear activation functions let me approximate my dataset?



Let $\sigma(x)$ be the sigmoid function,

$$\sigma(x) = \frac{e^x}{1 + e^x}$$

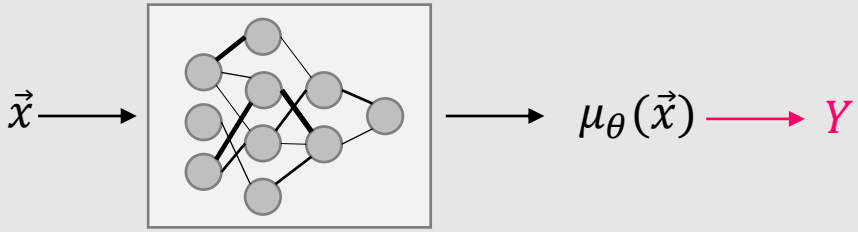


QUESTIONS FROM WEDNESDAY

How would we specify a probability model that uses several predictor variables?

$$Y \sim N(\mu_\theta(\vec{x}), \sigma^2)$$

where $\vec{x}$ is a tuple of predictor variables, $\vec{x} = (x_1, \ldots, x_d)$
and $\mu_\theta : \mathbb{R}^d \to \mathbb{R}$



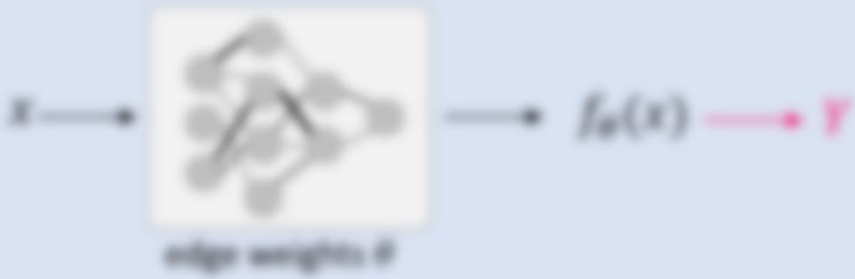
Supervised Learning

Data: $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

Labels: $y_1, y_2, \dots, y_n$

Task: fit the probability model $\Pr_Y(y : f_\theta(x))$

Training goal: MLE



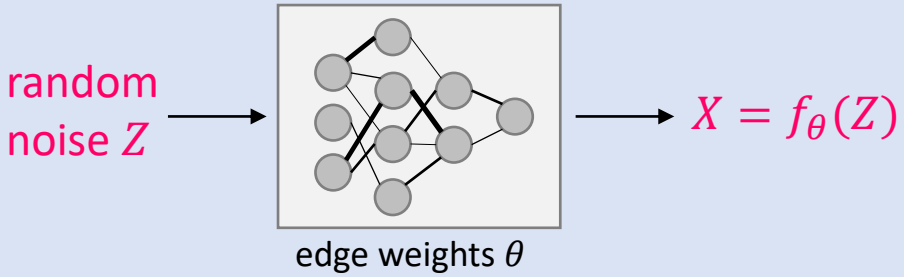
Generative Modelling

Data: $\{x_1, x_2, \dots, x_n\}$

Labels: n/a

Task: fit the probability model $\Pr_X(x ; \theta)$

Training goal: MLE



QUESTIONS FROM WEDNESDAY

I've read that all generative modelling is basically about Autoencoders. What does this have to do with probability models?

- To train a generative model, we write out the log likelihood of our dataset $[x_1, \dots, x_n]$ and find the parameters that maximize it:

$$\max_{\theta} \sum_i \log \Pr_X(x_i; \theta)$$

- This requires that we have a formula for $\Pr_X(x; \theta)$

CASE 1: There is a nice explicit expression (e.g. Transformers)

CASE 2: There is no closed-form expression (e.g. CNNs for image generation)

Both Autoencoders and GANs can be seen as ways to approximate $\Pr_X(x; \theta)$

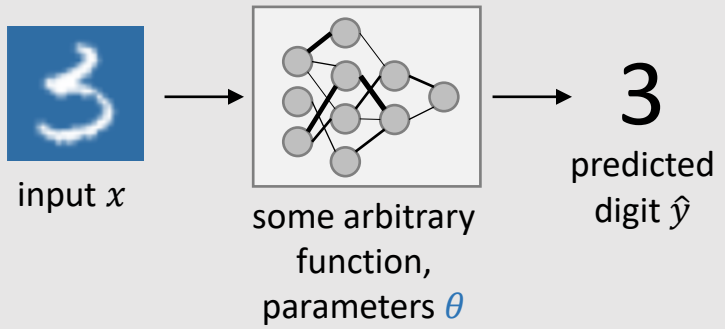
QUESTIONS FROM WEDNESDAY

In classification, the output is discrete. So how can gradient descent help?

Example (classification)

The MNIST dataset consists of pairs (x_i, y_i), where each record consists of $x_i \in \mathbb{R}^{28\times 28}$ an image of a handwritten digit and $y_i \in \{0,1,\ldots,9\}$ is its label.

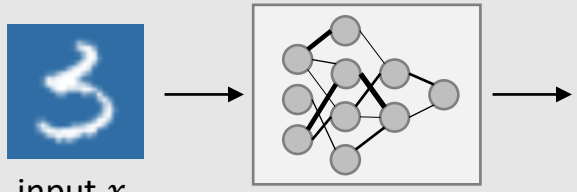
We'd like to predict the digit, given an image. How might we learn to do this?



Example (classification)

The MNIST dataset consists of pairs (x_i, y_i), where each record consists of $x_i \in \mathbb{R}^{28\times 28}$ an image of a handwritten digit and $y_i \in \{0,1,\ldots,9\}$ is its label.

Devise a probabilistic model to predict the label of a given input image, and fit it.



input x

some arbitrary function, parameters θ

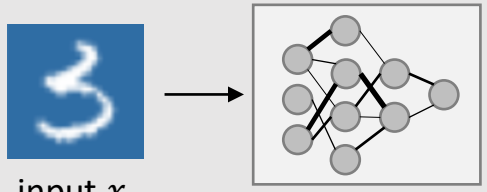
random output Y

What sort of PROBABILITY MODEL might we use for the response Y?

Example (classification)

The MNIST dataset consists of pairs (x_i, y_i), where each record consists of $x_i \in \mathbb{R}^{28\times 28}$ an image of a handwritten digit and $y_i \in \{0,1,\dots,9\}$ is its label.

Devise a probabilistic model to predict the label of a given input image, and fit it.



input x

some arbitrary function, parameters θ

$$\vec{p}(x) = [p_0, \dots, p_9] \longrightarrow Y \sim \mathrm{Cat}(\vec{p}(x))$$

How can we make sure that $\vec{p}$ is a valid probability vector?

(We need $p_i \in [0,1]$ for each i, and $\Sigma_i p_i = 1$.)

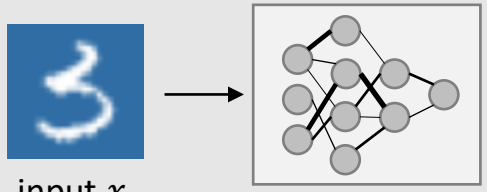
Example (classification)

The MNIST dataset consists of pairs (x_i, y_i), where each record consists of $x_i \in \mathbb{R}^{28\times 28}$ an image of a handwritten digit and $y_i \in \{0,1,\dots,9\}$ is its label.

Devise a probabilistic model to predict the label of a given input image, and fit it.

Softmax function:

$$p_k = \frac{e^{s_k}}{\Sigma_{\ell=0}^{9} e^{s_\ell}}$$



input x

some arbitrary function, parameters θ

$$\vec{s} \in \mathbb{R}^{10} \longrightarrow \vec{p} = \text{softmax}(\vec{s}) \longrightarrow Y \sim \text{Cat}(\vec{p})$$

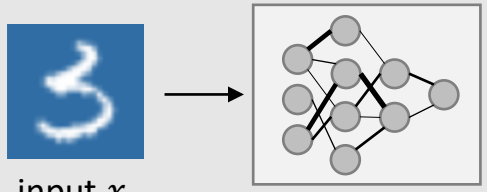
How should we fit the function parameters θ?

Example (classification)

The MNIST dataset consists of pairs (x_i, y_i), where each record consists of $x_i \in \mathbb{R}^{28\times 28}$ an image of a handwritten digit and $y_i \in \{0,1,\ldots,9\}$ is its label.

Devise a probabilistic model to predict the label of a given input image, and fit it.

Model for a single datapoint:



input x

some arbitrary function, parameters θ

$$\vec{s} \in \mathbb{R}^{10} \longrightarrow \vec{p} = \text{softmax}(\vec{s}) \longrightarrow Y \sim \text{Cat}(\vec{p})$$

Likelihood of a single datapoint y:

$$\Pr_Y(y; x, \theta) =$$

Log likelihood of the dataset:

$$\log \Pr(y_1, \ldots, y_n) =$$

… we end up with the famous "softmax cross-entropy loss function"