Instituto Superior Técnico

Deep Learning

2^{nd} Quarter - 2022/2023

Homework 1

Group 46:
Santiago Quintas - 93179
Alina Kling - 105208

Professor:
Mário Figueiredo

23^{st} December, 2022

Declaration of Contribution

We worked on all of the Questions together. In Question 1 we did together in a Discord Call, the other Questions we worked in separate and compared the results. In Question 2 Santiago was leading the process of development, while Alina did so in Question 3.

Question 1

1.

a)

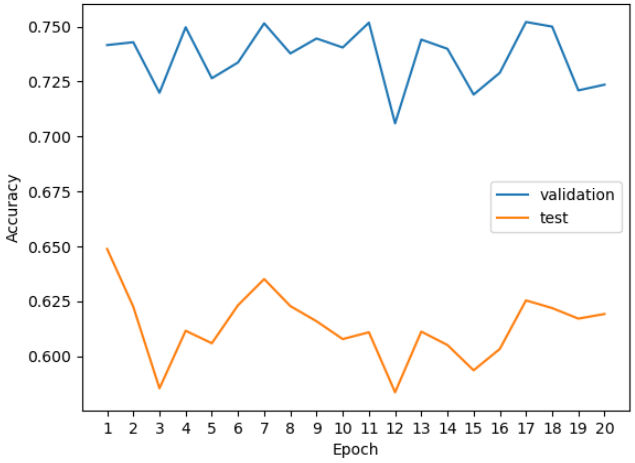


Figure 1: Accuracy of the perceptron as a function of the epoch number

b)

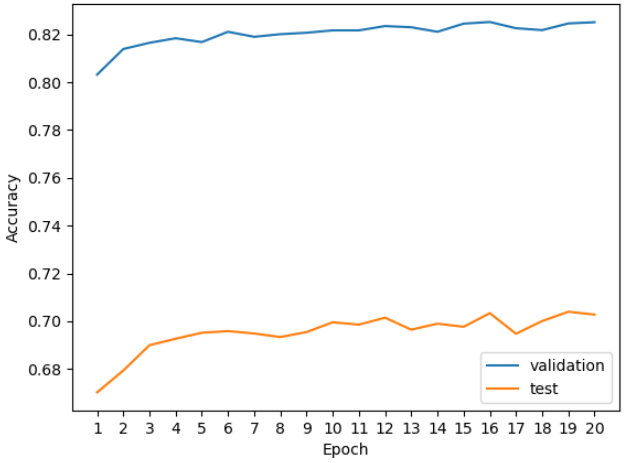


Figure 2: Accuracy of the logistic regression as a function of the epoch number

2.

a)

A multi layered perceptron, (mlp), allows the implementation of non-linear activation functions in its hidden layers, while a simple perceptron is limited to a linear activation function in its output. The main benefit of a mlp is having the capability to learn more complex patterns of data with non linear relationships while a simple perceptron is only useful in data with linear relationships. In this particular task our data is not linearly separable, the mlp will be able to overcome this issue with non linear activation functions.

This advantage goes away if the mlp is implemented with only linear activation functions, the mlp becomes a more complex equivalent of a simple perceptron, the output of a hidden layer is the input of the next one, and if they all possess linear activation functions then the output of the perceptron will be a linear relationship between the weights and biases of the previous layers, which removes the expressiveness necessary for learning more complex patterns.

b)

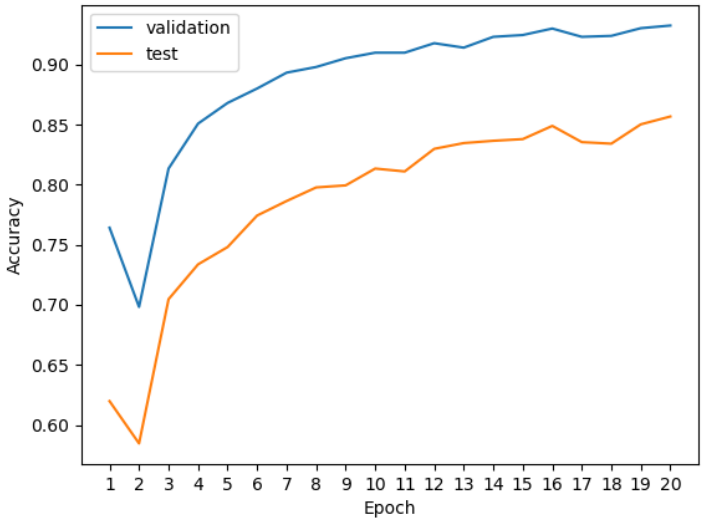


Figure 3: Accuracy of the multi layered perceptron (mlp) as a function of the epoch number

Question 2

1.

The default parameters used in the implementation of the logistic regression linear model were the following:

Number of Epochs	20
Learning Rate	0.01
Batch Size	1

Table 1: Default parameters 1

For different learning rates we achieved different accuracies, shown in the table 2. The best configuration, according to the accuracy test is with a learning rate of 0.001

Learning Rate	Final test accuracy
0.1	0.6133
0.01	0.6806
0.001	0.7019

Table 2: Accuracy test for different learning rates

Below is the plot of both the training loss and accuracy over the number of epochs of the changes for a learning rate of 0.001, which corresponds to our best configuration.

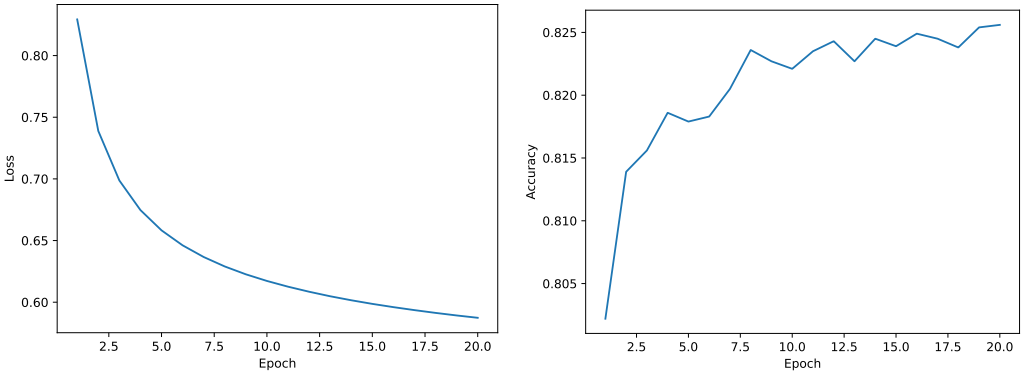


Figure 4: Training loss and accuracy of the logistic regression for a learning rate of 0.001

2.

The default parameters used in the implementation of the neural network with a single layer were the following:

Number of Epochs	20
Learning Rate	0.01
Hidden Size	100
Dropout	0.3
Batch Size	16
Activation	ReLU
Optimizer	SGD

Table 3: Default parameters 2

For different parameters the accuracy test ended with these results shown in table 4, keep in mind that only 1 parameter was changed at a time, the rest remained in their default setting. The best configuration was the one with a hidden size of 200.

Parameter	Value	Final test accuracy
Learning Rate	0.1	0.8714
Learning Rate	0.01	0.8593
Learning Rate	0.001	0.7450
Hidden Size	200	0.8811
Dropout	0.5	0.8413
Activation	tanh	0.8257

Table 4: Accuracy test for different parameters

Below is the plot of both the training loss and accuracy over the number of epochs for a hidden size of 200, which corresponds to our best configuration.

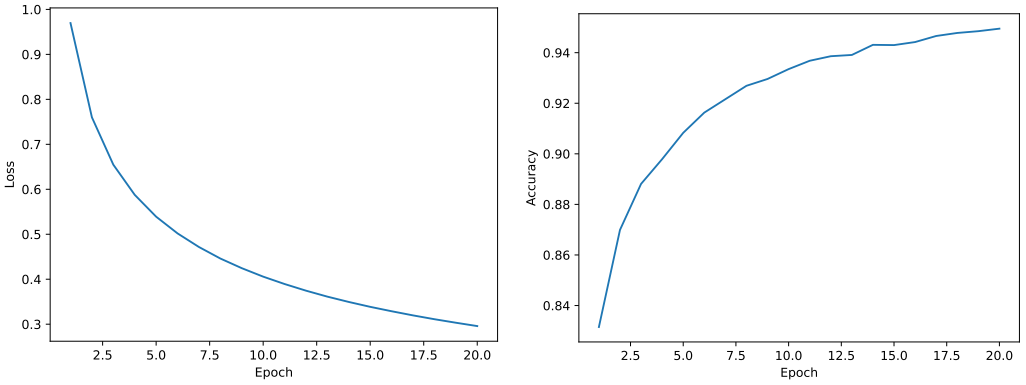


Figure 5: Training loss and accuracy of the mlp for a hidden size of 200

3.

Using the same default parameters as before we update the number of layers in our mlp and the results obtained are shown in the table 5, the best configuration was the one with 2 layers.

Number of Layers	Final test accuracy
1 (default)	0.8593
2	0.8698
3	0.8652

Table 5: Accuracy test for different learning rates

Below is the plot of both the training loss and accuracy over the number of epochs for a mlp with 2 layers, which corresponds to our best configuration.

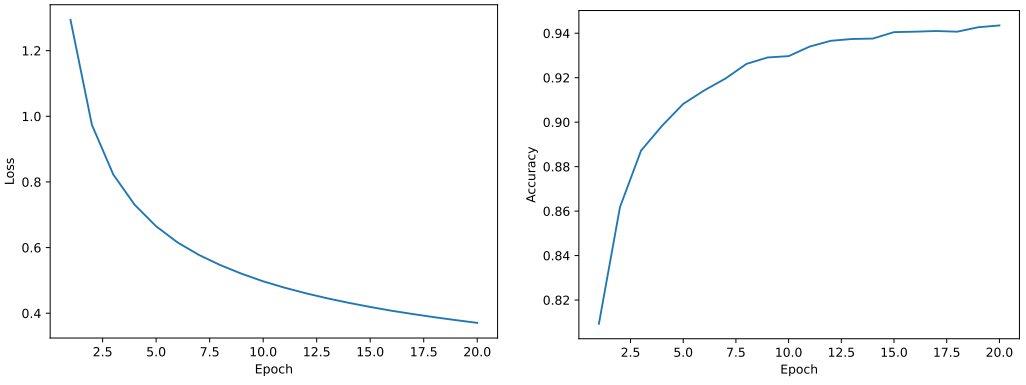


Figure 6: Training loss and accuracy of the mlp using 2 layers

Homework 1

Question 3 (30 points)

Multi-layer perceptron with quadratic activations. In this exercise, we will consider a feed-forward neural network with a single hidden layer and a quadratic activation function, $g(z) = z^2$. We will see under some assumptions, this choice of activation, unlike other popular activation functions such as tanh, sigmoid, or relu, can be tackled as a linear model via a reparametrization.

We assume a univariate regression task, where the predicted output $\hat{y} \in \mathbb{R}$ is given by $\hat{y} = \boldsymbol{v}^\top \boldsymbol{h}$, where $\boldsymbol{h} \in \mathbb{R}^K$ are internal representations, given by $\boldsymbol{h} = \boldsymbol{g}(\boldsymbol{W}\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^D$ is a vector of input variables, and $\Theta = (\boldsymbol{W}, \boldsymbol{v}) \in \mathbb{R}^{K \times D} \times \mathbb{R}^K$ are the model parameters.

- feed-forward NN: 1 single hidden layer
  quadratic activation function $g(z) = z^2$
  ↳ with assump. with reparametrization ≈ lin. model
- Assumptions: univariate regression task
  predicted output $\hat{y} \in \mathbb{R}$: $\hat{y} = v^T \vec{h}$
  internal representations $\vec{h} \in \mathbb{R}^K$: $\vec{h} = \vec{g}(\vec{W}\vec{x})$
  vector of input variables $\vec{x} \in \mathbb{R}^D$
  model parameters $\theta = (\vec{W}, \vec{v}) \in \mathbb{R}^{K \times D} \times \mathbb{R}^K$: $\theta = (\vec{W}, \vec{v})$

1. (10 points) Show that we can write $\boldsymbol{h} = \boldsymbol{A}_\Theta \boldsymbol{\phi}(\boldsymbol{x})$ for a certain feature transformation $\boldsymbol{\phi} : \mathbb{R}^D \to \mathbb{R}^{\frac{D(D+1)}{2}}$ independent of Θ and $\boldsymbol{A}_\Theta \in \mathbb{R}^{K \times \frac{D(D+1)}{2}}$. That is, $\boldsymbol{h}$ is a **linear transformation** of $\boldsymbol{\phi}(\boldsymbol{x})$. Determine the mapping $\boldsymbol{\phi}$ and the matrix $\boldsymbol{A}_\Theta$.

a) $h = A_\theta \Phi(x)$ for $\Phi : \mathbb{R}^D \to \mathbb{R}^{\frac{D(D+1)}{2}}$ independent of θ and $A_\theta \in \mathbb{R}^{K \times \frac{D(D+1)}{2}}$

Goal: $h = g(Wx) \doteq A_\theta \cdot \Phi(x)$

(matrix, depends only on θ) (vector $\frac{D(D+1)}{2}$, depends only on the input $\vec{x}$)

$$\underbrace{[1\ 2\ 3]}_{1\times 3} \cdot \underbrace{\begin{bmatrix} 2 & 1 & 3 \\ 3 & 3 & 2 \\ 4 & 1 & 2 \end{bmatrix}}_{3\times 3} = \begin{bmatrix} 1\cdot 2+2\cdot 3+3\cdot 4 \\ 1\cdot 1+2\cdot 3+3\cdot 1 \\ 1\cdot 3+2\cdot 2+3\cdot 2 \end{bmatrix} = \underbrace{\begin{bmatrix} 20 \\ 10 \\ 13 \end{bmatrix}}_{1\times 3}$$

(I) Use $g(z) = z^2$

$$g(Wx) = g\left(\sum_{i=1}^{D} W_{ij} x_i\right) = \left(\sum_{i=1}^{D} W_{ij} x_i\right)^2 = \sum_{l=1}^{D}\sum_{i=1}^{D} W_{ij} x_i W_{lj} x_l \qquad (W x_{ij})$$

→ Main idea: Show that the above sum can be rewritten as a multiplication

of a matrix (pure of θ) and a pure vector of x

(II) To understand summation: Example with $D=3$ and $K=2$

$\phi : \mathbb{R}^D \to \mathbb{R}^{D(D+1)/2}$ (with same index with another index; to eliminate doubles!)

(number of perturbations for $D=3$: $\mathbb{R}^3 \to \mathbb{R}^{\frac{3(4)}{2}} = \mathbb{R}^6$ i.e. 11, 13, 23, ...)

$W \in \mathbb{R}^{K \times D} \to W \triangleq 2 \times 3$ matrix

$x \in \mathbb{R}^D \to x : 3 \times 1$ vector

calculate $(Wx)^2$

$$Wx = \begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_i w_{1i} \\ x_i w_{2i} \end{pmatrix} \text{ with } i \in \{1,2,3\}$$

$$(Wx)^2 = \begin{pmatrix} (x_i w_{1i})^2 \\ (x_i w_{2i})^2 \end{pmatrix} = \begin{pmatrix} (x_1 w_{11} + x_2 w_{12} + x_3 w_{13})^2 \\ (x_1 w_{21} + x_2 w_{21} + x_3 w_{23})^2 \end{pmatrix}$$

$$= \begin{pmatrix} (x_1 w_{11})^2 + (x_2 w_{12})^2 + (x_3 w_{13})^2 + 2x_1x_2 w_{11}w_{12} + 2x_1x_3 w_{11}w_{13} + 2x_2x_3 w_{12}w_{13} \\ (x_1 w_{21})^2 + (x_2 w_{22})^2 + (x_3 w_{23})^2 + 2x_1x_2 w_{21}w_{22} + 2x_1x_3 w_{21}w_{23} + 2x_2x_3 w_{22}w_{23} \end{pmatrix}$$

$\Rightarrow \phi(x) : \mathbb{R}^D \to \mathbb{R}^{D(D+1)/2}$

$(x_1, \dots, x_D) \to (x_1^2, x_2^2, \dots, x_1x_2, x_1x_3, \dots)$ (D-elements; $D(D+1)/2$-elements)

Try to rewrite this now to (Ws) $\begin{pmatrix} x_1^2 \\ \vdots \\ x_1x_2 \\ x_2x_3 \\ \vdots \end{pmatrix}$

$$\begin{matrix} j=1 \\ j=2 \end{matrix} \begin{pmatrix} w_{11}^2 & w_{12}^2 & w_{13}^2 & 2w_{11}w_{12} & 2w_{13}w_{11} & 2w_{12}w_{13} \\ w_{21}^2 & w_{22}^2 & w_{23}^2 & 2w_{21}w_{22} & 2w_{21}w_{23} & 2w_{32}w_{23} \end{pmatrix} \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \\ x_1x_2 \\ x_1x_3 \\ x_2x_3 \end{pmatrix}$$

$A\theta$: depends only on parameters W — $\phi(x)$: depends only on input x

Can write the $\phi(x)$ as:

$$\phi(x) = [x_1x_1, x_1x_2, \dots, x_2x_2, x_2x_3, \dots, x_{D-1}x_D, x_D^2]$$

↖ all possible combinations of $x_i x_j$

(or $\sum_{j=1}^{D} \sum_{i=1}^{D} x_i x_j$)

III) Write down generilized version of II)

h for each row j of the h rows $\sum_{i=1}^{D}\sum_{\ell=1}^{D}(w_{je}w_{ji})(x_i x_\ell) = \sum_{j=1}^{D} w_{ij}^2 x_i^2 + 2\sum_{i=1}^{D}\sum_{\ell=i+1}^{D} x_i x_\ell w_{je} w_{ji}$

$(w_{je}w_{ji}) =: A\theta_j$; $(x_i x_\ell) =: \phi_j$; $\sum_{j=1}^{D} w_{ij}^2 x_i^2$: D-terms (quadratic); $2\sum_{i=1}^{D}\sum_{\ell=i+1}^{D} x_i x_\ell w_{je} w_{ji}$: mixed terms $\frac{D(D-1)}{2}$

redefine the indecies of the sums and re-write them as one sum over m

with $n = \{ij\}$ all combinations of ij (without repetitions)

↳ D values, D values

$\Rightarrow m = \{1, \dots, \frac{D(D+1)}{2}\}$

for the jth row of h $\Rightarrow h_j = \sum_{m=1}^{D(D+1)/2} A\theta_{jm}\phi_m \Rightarrow A\theta\,\phi(x)$ ✓

$\Rightarrow$ so it was shown, that h is a linear transformation $\phi(x)$ for the feature transformation with mapping

$\phi[x_1, \dots, x_D] \rightarrow \phi[x_1x_1 \mid x_1x_2 \mid \dots \mid x_2x_2 \mid x_2x_3 \mid \dots \mid x_{D-1}x_D \mid x_D^2]$

and that the matrix $A\theta$ has the j rows $\sum_{i=1}^{D}\sum_{\ell=1}^{D} w_{je}w_{ji}$

III Alternative zu II above in Matrix-Form

$$h = g(wx) = (wx)^2 = \left(\sum_{j=1}^{D} w_{ij}x_j\right)^2 = \begin{pmatrix} \left(\sum_{j=1}^{D} w_{1j}x_j\right)^2 \\ \vdots \\ \left(\sum_{j=1}^{D} w_{nj}x_j\right)^2 \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{D} w_{1j}^2 x_1^2 + 2\sum_{i=1}^{D}\sum_{\ell=i+1}^{D} x_i x_\ell w_{1\ell} w_{1i} \\ \vdots \\ \sum_{j=1}^{D} w_{nj}^2 x_i^2 + 2\sum_{i=1}^{D}\sum_{\ell=i+1}^{D} x_i x_\ell w_{n\ell} w_{ni} \end{pmatrix}$$

$\sum_{j=1}^{D} w_{ij}^2 x_i^2$: D-Terms; $2\sum_{i=1}^{D}\sum_{\ell=i+1}^{D} x_i x_\ell w_{n\ell} w_{ni}$: $\frac{D(D-1)}{2}$ Terms

$\Rightarrow \frac{D(D+1)}{2}$ Terms

$$= \begin{pmatrix} w_{11}^2 & \dots & w_{1D}^2 & w_{11}w_{12} & \dots & w_{1(D-1)}w_{1D} \\ \vdots & & \vdots & \vdots & & \vdots \\ w_{n1}^2 & \dots & w_{nD}^2 & w_{n1}w_{n2} & \dots & w_{n(D-1)}w_{nD} \end{pmatrix} \cdot \begin{pmatrix} x_1^2 \\ \vdots \\ x_D^2 \\ 2x_1x_2 \\ \vdots \\ 2x_{D-1}x_D \end{pmatrix}$$

$=: A\theta \in \mathbb{R}^{n \times D(D+1)/2}$; $=: \phi(x) \in \mathbb{R}^{\frac{D(D+1)}{2}}$

$\Rightarrow h = A\theta \cdot \phi(x)$ (so its linear + we can separate)

2. (5 points) a) Based on the previous claim, show that $\hat{y}$ is also a linear transformation of $\phi(\boldsymbol{x})$, i.e., we can write $\hat{y}(\boldsymbol{x}; \boldsymbol{c}_\Theta) = \boldsymbol{c}_\Theta^\top \phi(\boldsymbol{x})$ for some $\boldsymbol{c}_\Theta \in \mathbb{R}^{\frac{D(D+1)}{2}}$ b) Does this mean this is a linear model in terms of the original parameters Θ?

a)

Idea:

predicted output $\hat{y} \in \mathbb{R}$: $\hat{y} = v^T h$

internal representations $\vec{h} \in \mathbb{R}^K$: $\vec{h} = \vec{g}(\vec{w}\vec{x})$

vector of input variables $\vec{x} \in \mathbb{R}^D$

model parameters $\theta = (\vec{w}, \vec{v}) \in \mathbb{R}^{K \times D} \times \mathbb{R}^K$: $\theta = (\vec{w}, \vec{v})$

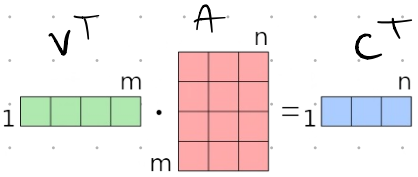
$\Rightarrow$ so should be straight forward: if h from above linear can just plug it in?

(I) Use definition of $\hat{y}$ and plug h from 3.1. in

(h from 3.1: $h = A_\theta \cdot \phi(x)$ with A_θ = matrix with elements $A_{ij} = w_{ij}$ and $\phi(x)$ = vector of monomials of x up to 2nd order

$\Rightarrow \hat{y} = v^T h = v^T A_\theta \cdot \phi(x)$

$\Rightarrow c_\theta^T = v^T A_\theta$



(II) Therefore we can write $\hat{y}$ as a linear transformation of phi(x)

b) Model = linear in terms of the original parameters $\theta = (\vec{w}, \vec{v}) \in \mathbb{R}^{K \times D} \times \mathbb{R}^K$?

NOT necessarily, cause we can see in 3.1 (II) that A_θ and therefore also c_θ^T which depends on A_θ is not linear in the parameters in the original parameter w.

(because the weights are multiplied with each other (e.g. squared) and mulitplied with v)

3. (10 points) Assume $K \geq D$. Show that for any real vector $\boldsymbol{c} \in \mathbb{R}^{\frac{D(D+1)}{2}}$ there is a choice of the original parameters $\Theta = (\boldsymbol{W}, \boldsymbol{v})$ such that $\boldsymbol{c}_\Theta = \boldsymbol{c}$. That is, **we can equivalently parametrize the model with $\boldsymbol{c}_\Theta$ instead of Θ.** Does this mean this is a linear model in terms of $\boldsymbol{c}_\Theta$? Show that an equivalent parametrization might not exist if $K < D$.

https://cs229.stanford.edu/section/cs229-linalg.pdf

https://www.deeplearningbook.org/contents/linear_algebra.html

single answer

In Q2, you show that for any Θ there is a corresponding c that makes the network equivalent to a linear model.

In Q3, the goal is to show the reverse direction -- that for any given c there is a corresponding Θ in the original model so that the two models are equivalent. This is the most difficult of that group of questions. The requirement of $K \geq D$ will arise if you are on the right path.

What should be the bridge between that conclusion for D(D+1)/2 and for D? I've worked out a potential proof but I ended up reaching the same conclusion as my colleague. Thanks in advance.

helpful! | 0

Mário Figueiredo 2 days ago

Please keep in mind that the relationship between the parameters Θ and c is **not** linear. Thinking in terms of the number of equations versus the number of unknowns is only valid for **linear** systems of equations.

good comment | 0

- The relationship between y and x is certainly non-linear.
- In Q3.1 and Q3.2 you prove that there is a transformation $\phi : \mathbb{R}^D \to \mathbb{R}^{\frac{D(D+1)}{2}}$ such that y is a linear function of $\phi(x)$, that is, it can be written as $y = c_\Theta^T \phi(x)$.
- The parameter c_Θ that appears in the previous paragraph is a non-linear function of the original network parameters $\Theta = (W, v)$, that is, we can obtain it via some non-linear function $\psi : \mathbb{R}^{K \times D} \times \mathbb{R}^K \to \mathbb{R}^{\frac{D(D+1)}{2}}$, such that $c_\Theta = \psi(W, v)$.
- Question Q3.3 can be stated as follows: **(1)** show that if $K \geq D$, then, for any $c \in \mathbb{R}^{\frac{D(D+1)}{2}}$, there exist a pair (W, v) such that $c = \psi(W, v)$. **(2)** Show that if $K < D$, there may exist some $c \in \mathbb{R}^{\frac{D(D+1)}{2}}$, such that there is **no** (W, v) that satisfies $c = \psi(W, v)$.

I hope this helps to clarify the question.

$c_\Theta^T = v^T A_\Theta \quad | ^T$

$(c_\Theta^T)^T = (v^T A_\Theta)^T$

$c_\Theta = A_\Theta^T v := c$ and $c = \psi(W, v)$

$\frac{D(D+1)}{2} \times K \quad K \quad \frac{D(D+1)}{2}$

can I divide the vector c in Matrix times vector? → with right Dimensions?

↳ hmmmm??? ∴

model parameters $\theta = (\vec{W}, \vec{v}) \in \mathbb{R}^{K \times D} \times \mathbb{R}^K : \theta = (\vec{W}, \vec{v})$

$\hat{y}(x; c_\Theta) = c_\Theta^T \phi(x)$ for some $c_\Theta \in \mathbb{R}^{\frac{D(D+1)}{2}}$

$y = v^T h = v^T A_\theta \cdot \phi(x)$

$c_\theta^T = v^T A_\theta$

- Look at dimensions:

internal representations $\vec{h} \in \mathbb{R}^K$: $\vec{h} = \vec{g}(\vec{W}\vec{x})$

vector of input variables $\vec{x} \in \mathbb{R}^D$

model parameters $\theta = (\vec{W}, \vec{v}) \in \mathbb{R}^{K \times D} \times \mathbb{R}^K : \theta = (\vec{W}, \vec{v})$

if $K \geq D$: more internal representations than input variables

then ? :(

I think we need this to ensure (⟹) [illegible] exist?

with $Ax = \sum x_i A_{:,i}$

no, but it's not linear?

• Look at definition of linearity:

Deep-learning-book:

In order for the system $\boldsymbol{Ax} = \boldsymbol{b}$ to have a solution for all values of $\boldsymbol{b} \in \mathbb{R}^m$, we therefore require that the column space of $\boldsymbol{A}$ be all of $\mathbb{R}^m$. If any point in $\mathbb{R}^m$ is excluded from the column space, that point is a potential value of $\boldsymbol{b}$ that has no solution. The requirement that the column space of $\boldsymbol{A}$ be all of $\mathbb{R}^m$ implies immediately that $\boldsymbol{A}$ must have at least m columns, that is, $n \geq m$. Otherwise, the dimensionality of the column space would be less than m. For example, consider a 3×2 matrix. The target $\boldsymbol{b}$ is 3-D, but $\boldsymbol{x}$ is only 2-D, so modifying the value of $\boldsymbol{x}$ at best enables us to trace out a 2-D plane within $\mathbb{R}^3$. The equation has a solution if and only if $\boldsymbol{b}$ lies on that plane.

Having $n \geq m$ is only a necessary condition for every point to have a solution. It is not a sufficient condition, because it is possible for some of the columns to be redundant. Consider a 2×2 matrix where both of the columns are identical. This has the same column space as a 2×1 matrix containing only one copy of the replicated column. In other words, the column space is still just a line and fails to encompass all of $\mathbb{R}^2$, even though there are two columns.

Formally, this kind of redundancy is known as **linear dependence**. A set of vectors is **linearly independent** if no vector in the set is a linear combination of the other vectors. If we add a vector to a set that is a linear combination of the other vectors in the set, the new vector does not add any points to the set's span. This means that for the column space of the matrix to encompass all of $\mathbb{R}^m$, the matrix must contain at least one set of m linearly independent columns. This condition is both necessary and sufficient for equation 2.11 to have a solution for every value of $\boldsymbol{b}$. Note that the requirement is for a set to have exactly m linearly independent columns, not at least m. No set of m-dimensional vectors can have more than m mutually linearly independent columns, but a matrix with more than m columns may have more than one such set.

4. (5 points) Suppose we are given training data $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ with $N > \frac{D(D+1)}{2}$ and that we want to minimize the squared loss

$$L(\boldsymbol{c}_\Theta; \mathcal{D}) = \frac{1}{2}\sum_{n=1}^{N}(\hat{y}_n(\boldsymbol{x}_n; \boldsymbol{c}_\Theta) - y_n)^2.$$

Let the matrix $\boldsymbol{X} \in \mathbb{R}^{N \times \frac{D(D+1)}{2}}$ have $\boldsymbol{\phi}(\boldsymbol{x}_n)$ as rows and assume that $\boldsymbol{X}$ has full column-rank. Can we find a closed form solution $\hat{\boldsymbol{c}}_\Theta$? Comment on the fact that global minimization is usually intractable for feedforward neural networks – what makes our problem special?

(I) What is a closed solution: lecture

As discribed in lecture 2 in slide 16 the closed form of a model, that is linear w.r.t the model parameters w: $\hat{y} = w^T \phi(x)$

with the feature vector $\phi(x)$ (e.g. $\phi(x) = [1, x, x^2, \dots x^D]^T \in \mathbb{R}^{D+1}$ ^bias term

is given as

$$\hat{w} = (X^T X)^{-1} X^T y, \text{ with } X = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_n)^T \\ \vdots \\ \phi(x_N)^T \end{bmatrix}, \; y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ \vdots \\ y_N \end{bmatrix}.$$

This solution can be found by minimizing the squared loss with

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2 \quad \text{with } \hat{y} = w^T \phi(x)$$

with respect to w.

In order to do so write L in matrix-vector notation:

$$L(y, \hat{y}) = \sum_{n=1}^{N} L(y_n, \hat{y}_n) = \frac{1}{2}\sum_{n=1}^{N}(y_n - w^T \phi(x_n))$$

$$= \|y - Xw\|^2$$

Find minimum ($\min_{w \in \mathbb{R}^{D+1}} \frac{1}{2}\sum_{n=1}^{N}(y_n - w^T \phi(x_n))^2$):

$$0 = \nabla_w \|y - Xw\|^2$$

$$= \nabla_w \|w^T X^T X w - 2 w^T X^T y + \|y\|^2$$

$$0 = 2X^T X w - 2X^T y$$

$\Rightarrow$ closed form solution is $\hat{w} = (X^T X)^{-1} X^T y$

(II) Apply this to our example

From (I) we can conclude, that it should be also possible to find a closed form solution $\hat{c}_\theta$ should be found, since

$\hat{y}$ is a linear model in terms of the parameters c_θ

as shown in Question 3.2: $\hat{y}(x; c_\theta) = c_\theta^T \phi(x)$

with $X = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_{n\times 1} \end{bmatrix}$ and $y = \begin{bmatrix} y_1 \\ y_n \\ y_N \end{bmatrix}$ and $\hat{y}(x_n; c_\theta) = c_\theta^T \phi(x_n)$

and the Loss function

$$L(c_\Theta; \mathcal{D}) = \frac{1}{2} \sum_{n=1}^{N} (\hat{y}_n(x_n; c_\Theta) - y_n)^2 = \frac{1}{2} \| X \cdot c_\theta - y \|^2$$

the closed form solution should be (proof in appendix :)

$$\hat{c}_\theta = (X^T X)^{-1} X^T y$$

according to (I)

↖ we can get this from $\nabla_{c_\theta}(\mathcal{L}(c_\theta; \mathcal{D})) = 0$ and its okay to change the order to free c_θ since the inverse of $X^T X$ exist because X has full rank!

(III) Global minimization in feed forward NNs

In general: Loss functions of feed forward NNs = non-convex

⇒ local minima exist

⇒ but global minima are intracable

our case: is special, because we apply the quadratic activation $g(z) = z^2$ and as shown in questions 3.1 and 3.2 with certain assumptions a feature transformation can be used to write the model as a linear model $\hat{y}(x; c_\theta)$ in terms of the parameters $c_\theta \in \mathbb{R}^{D(D+1)/2}$ (as said in 3.3.2 :)

⇒ squared loss $\mathcal{L}_\theta(c_\theta; \mathcal{D})$ is convex

⇒ closed form solution exists

Proof for closed form in 3.4

$c\theta = []$ vectors $y = []$

$X = [\,]$ Matrix

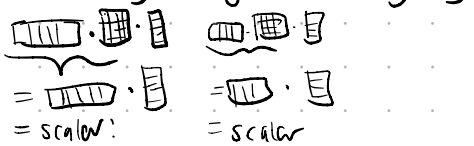
$\|X\theta - y\|$

Norm Definition: $\|A - B\| = (A-B)^T (A-B)$

$= (X\theta - y)^T (X\theta - y)$

$= (\theta^T X^T - y^T)(X\theta - y)$ — Transpose of product: $(AB)^T = B^T A^T$.

$= \theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y$



= scalar! = scalar

⇒ am allowed to write $2y^T X\theta$

$= \theta^T X^T X\theta - 2y^T X\theta + y^T y$ (or $-2\theta^T X^T y$) — Definition norm

quadratic

$\nabla_\theta \|X\theta - y\| = 2X^T X\theta - 2y^T X + 0 = 0 \quad | \cdot \frac{1}{2}$

$\theta = (X^T X)^{-1} y^T X$

or

$\theta = (X^T X)^{-1} X^T y$

Slides for 3.4

Linear Regression

Often, linear dependency of $\hat{y}$ on x is a bad/simplistic assumption

More general model: $\hat{y} = w^T \phi(x)$, where $\phi(x)$ is a feature vector

- e.g. $\phi(x) = [1, x, x^2, \ldots, x^D]^T \in \mathbb{R}^{D+1}$ (monomials degree $\leq D$)
- the bias term b is captured by the constant feature $\phi_0(x) = 1$

Fit/learn w by solving $\min_{w \in \mathbb{R}^{D+1}} \sum_{n=1}^{N} (y_n - w^T \phi(x_n))^2$

- Closed-form solution:

$$\hat{w} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}, \text{ with } \boldsymbol{X} = \begin{bmatrix} \phi(x_1)^T \\ \vdots \\ \phi(x_n)^T \\ \vdots \\ \phi(x_N)^T \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ \vdots \\ y_N \end{bmatrix}.$$

Still called linear regression – linear w.r.t. the model parameters w.

One-Slide Proof

Write the objective function in matrix-vector notation:

$$\sum_{n=1}^{N} (y_n - w^T \phi(x_n))^2 = \|\boldsymbol{y} - \boldsymbol{X}w\|^2$$

Equate the gradient to zero and solve the resulting equation:

$$\begin{aligned} 0 &= \nabla_w \|\boldsymbol{y} - \boldsymbol{X}w\|^2 \\ &= \nabla_w \left(w^T \boldsymbol{X}^T \boldsymbol{X} w - 2w^T \boldsymbol{X}^T \boldsymbol{y} + \|\boldsymbol{y}\|^2 \right) \\ &= 2\boldsymbol{X}^T \boldsymbol{X} w - 2\boldsymbol{X}^T \boldsymbol{y} \end{aligned}$$

Therefore

$$\hat{w} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}.$$

Squared Loss Function

Linear regression with least squares criterion corresponds to a loss function

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2, \quad \text{where } \hat{y} = w^T \phi(x).$$

This is called the squared loss.

The model is fit to the training data by minimizing the loss function:

$$\hat{w} = \arg\min_w \sum_{n=1}^{N} L(y_n - \hat{y}_n) = \arg\min_w \frac{1}{2} \sum_{n=1}^{N} (y_n - w^T \phi(x))^2$$

(the factor $1/2$ is irrelevant but convenient)

More later.