Imperial College London

May 11th, 2020

Recurrent Networks and LSTMs

Olivier Dubrule and Navjot Kukreja

1

Imperial College London

Objectives of the Presentation

- Present the basic model behind Recurrent Neural Networks (RNNs)
- Introduce Long Short Term-Memory (LSTM) Networks

2

Imperial College London

Examples of Sequential Data

- Text considered as a sequence of words or characters
- Continuous parameter which is a function of time (ie stock price)
- Sequence of images in a video-clip
- Sequence of labels in a Genome sequence
- Vertical sequences of lithologies in the subsurface
- …

3

Imperial College London

Examples of Applications related to Sequential Data

- Text Generation “in the style of”
- Provide ”Sentiment analysis” on a piece of text
- Automatic “Speech-to-Text” as in MS Teams
- Automatic Foreign language translation
- Prediction of stock price on the basis of historical data
- Label sequence of images in a video-clip

4

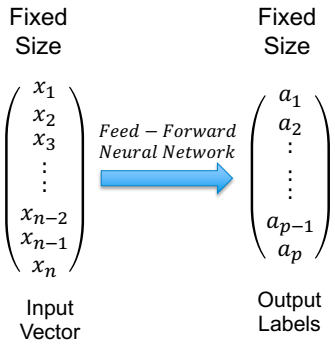
Imperial College London

A Simple Way to See Supervised Neural Networks

Suppose we have m pairs of data.
Each pair is composed of a vector of dimension n and a vector of dimension p (the labels).

A neural network is simply a function that maps any vector of dimension n into a (discrete or continuous) vector of dimension p.

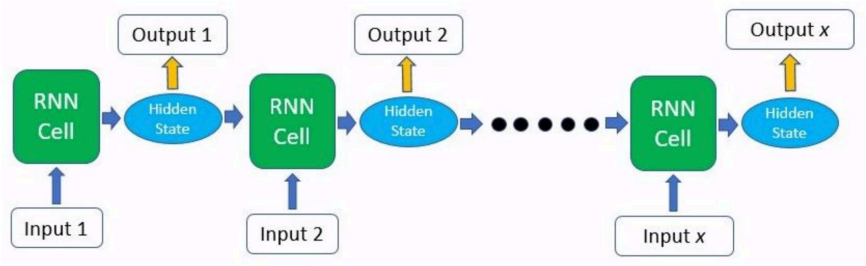
In order to calculate the parameters of this function, we train the parameters of the neural network by back-propagation using the m pairs of data as Training Set.



5

Imperial College London

Basic Recurrent Network Structure



Example:
Based on a (usually long) text used for Training, we wish to generate "similar" text on a character by character basis.

https://blog.floydhub.com/a-beginners-guide-on-recurrent-neural-networks-with-pytorch/

6

Imperial College London

A Simplistic Character Sequence Example (1)

Suppose the Training Text is: *Hello World!*

What is the Dictionary associated with this Training text?

If we ignore capital letters the Dictionary contains 9 characters: (d, e, h, l, o, r, w, , !)

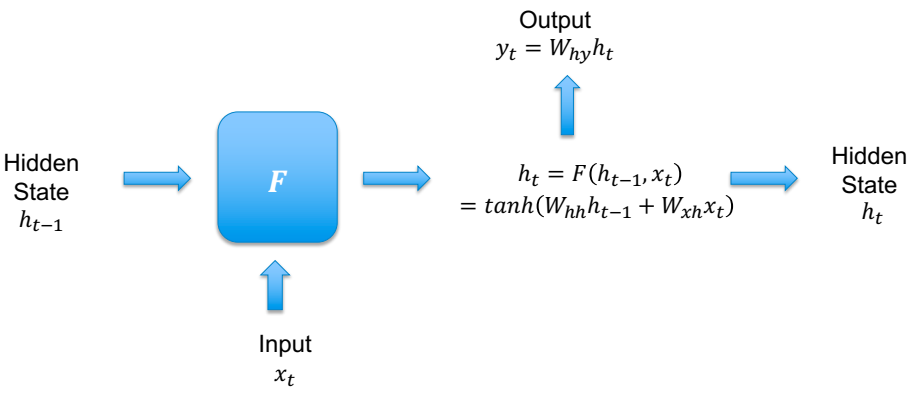
Each character is coded using one-hot encoding:

$$h=\begin{pmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix} \quad !=\begin{pmatrix}0\\0\\0\\0\\0\\0\\0\\0\\1\end{pmatrix} \quad \ldots..$$

7

Imperial College London

The Basic RNN Cell for time t



https://blog.floydhub.com/a-beginners-guide-on-recurrent-neural-networks-with-pytorch/

8

Imperial College London

A Simplistic Character Sequence Example (2)

TRAINING PROCESS (1) (assume that the hidden vector h_t is of dimension 5)

$$x_1 = \begin{pmatrix}0\\0\\1\\0\\0\\0\\0\\0\\0\end{pmatrix}$$ (x_1 codes the first letter "h" of the Training text)

$$h_0 = \begin{pmatrix}0\\0\\0\\0\\0\end{pmatrix}$$

$$h_1 = F(h_0, x_1) = tanh(W_{hh}h_0 + W_{xh}x_1)$$

W_{hh}: 5x5 matrix; W_{xh}: 5x9 matrix

$$h_1 = tanh\begin{pmatrix}w_{xh}^{1,3}\\w_{xh}^{2,3}\\w_{xh}^{3,3}\\w_{xh}^{4,3}\\w_{xh}^{5,3}\end{pmatrix}$$

$$y_1 = W_{hy}h_1$$

W_{hy}: 9x5 matrix

9

Imperial College London

A Simplistic Character Sequence Example (2)

TRAINING PROCESS (2)

x_2 is the hot-encoder vector associated with the second letter "e" of Training text

$$y_1 = W_{hy}h_1 \Rightarrow Cross - Entropy\,[Softmax(y_1), x_2]$$

Loss Function for first pair of letters

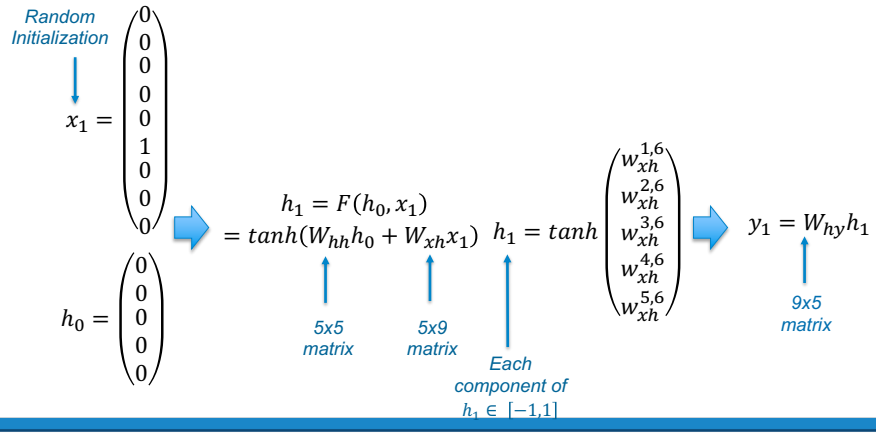
…then calculate Loss Function for second pair of letters by same process as previous slide, but starting with vectors x_2 and h_1, and add this Loss Function to the previous one, until end of current character sequence.

10

Imperial College London

A Simplistic Character Sequence Example (2)

TEXT GENERATION PHASE (1)



11

Imperial College London

A Simplistic Character Sequence Example (2)

TEXT GENERATION PHASE (2)

$y_1 = W_{hy}h_1$ ➡ Calculate $Softmax(y_1)$ ➡ Sample x_2 using the Softmax probabilities

…then generate x_3 using same process as previous slide with x_2 and h_1 as input

https://www.youtube.com/watch?v=6niqTuYFZLQ

12

Imperial College London

A Simplistic Character Sequence Example (3)

Training Workflow:

For a sequence of p (say $p = 20$) characters:

- Take first character x_1, combine with current hidden state h_0 (the size of h_0 is a hyper-parameter) to derive h_1, calculate associated output vector y_1 of size equal to vocabulary, transform y_1 into Softmax vector, calculate cross-entropy with second character x_2 of sequence.
- Take this second character as input to re-do what we just did with the first character and again calculate loss function using the value of the third character of the sequence, and so on until we reach the end of the sequence.
- Add up all the above loss functions.
- Do back-propagation to calculate gradients according to each trainable parameter.
- Modify parameters by gradient descent

Move to next sequence of p characters, until end of Training Set is reached.

13

Imperial College London

A Simplistic Character Sequence Example (4)

Text Generation Workflow:

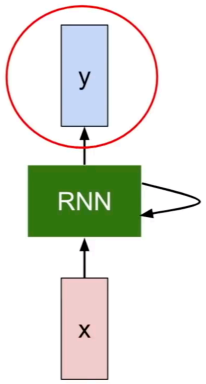
Generate a new sequence of n characters:

Randomly sample first character x_1, combine with initial null state vector value h_0 to derive new vector h_1, calculate associated output vector y_1 of size equal to vocabulary size, transform y_1 into Softmax vector, sample new character x_2 based on Softmax probability, and so on, until the required number of characters have been generated.

14

Imperial College London

Synthetic Notation for a Recurrent Network



y is a vector of output values y_t at different time steps

Contains the hidden state vector h_t which is fed back to the next sequence

x is a vector of input values x_t at different time steps

https://www.youtube.com/watch?v=6niqTuYFZLQ

15

Imperial College London

A Simple Language Processing Python Code

https://gist.github.com/karpathy/d4dee566867f8291f086

Learning to write Shakespeare plays

at first:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt    h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

After many iterations

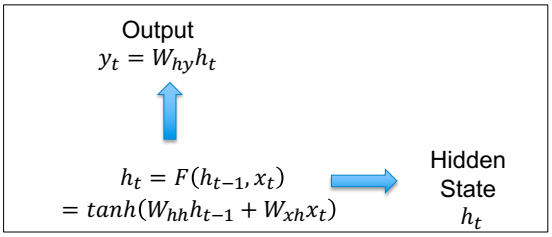
```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```

Andrej Karpathy's Blog: *The Unreasonable Effectiveness of Recurrent Neural Networks*

16

Imperial College London

A Summary



The weight matrices $\boldsymbol{W_{hh}}$, $\boldsymbol{W_{xh}}$ and $\boldsymbol{W_{hy}}$ contain the trainable parameters and are the same for each time t .

The hidden vector $\boldsymbol{h_{t-1}}$, the size of which is a hyperparameter, changes with t and contains recent historical information about the $\boldsymbol{x_t}$ sequence before t.

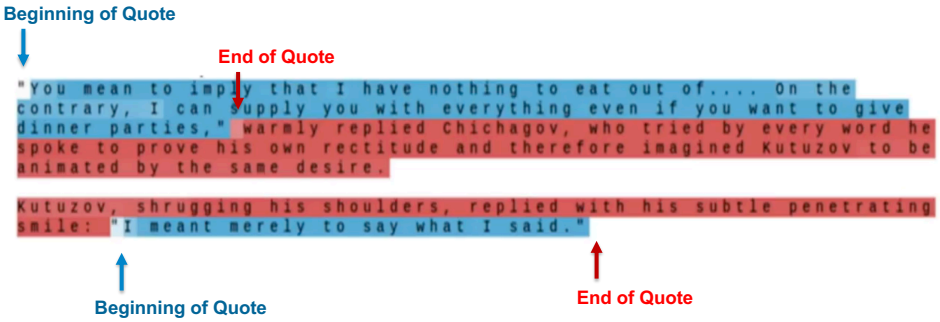
$\boldsymbol{x_t}$ is the new data at time t which is combined with $\boldsymbol{h_{t-1}}$ to predict $\boldsymbol{x_{t+1}}$.

https://www.youtube.com/watch?v=6niqTuYFZLQ

17

Imperial College London

Visualizing one coordinate of h_t as a function of t

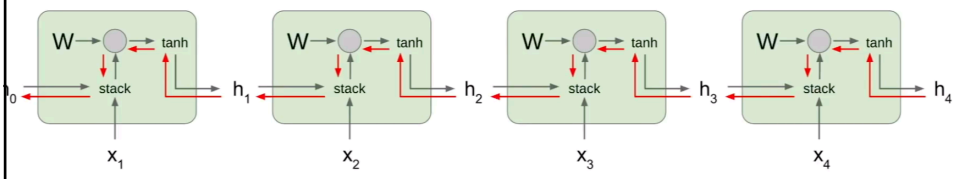


From Karpathy, Johnson and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016

18

Imperial College London

Back-Propagation through Recurrent Neural Network



$$h_t = tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$
$$= tanh\left((W_{hh}\ W_{xh})\begin{pmatrix}h_{t-1}\\x_t\end{pmatrix}\right)$$
$$= tanh\left(W\begin{pmatrix}h_{t-1}\\x_t\end{pmatrix}\right)$$

Computing gradients involves many factors of W (or W transposed)

Largest singular value > 1: **Exploding gradients**

Largest singular value < 1: **Vanishing gradients**

From Stanford University: https://www.youtube.com/watch?v=6niqTuYFZLQ

19

Imperial College London

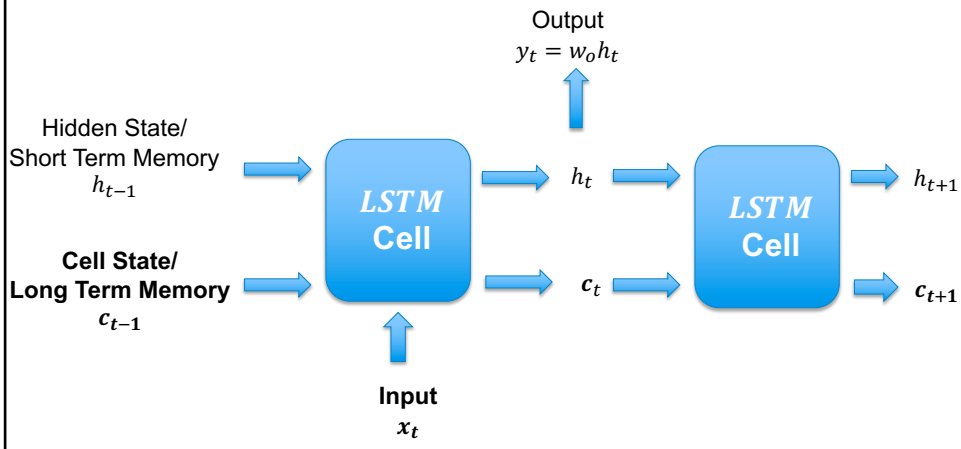
Issues with Recurrent Neural Networks

- Gradients may tend to Explode or Vanish
- Hidden state only covers Short-Term Memory

20

Imperial College London

The Basic LSTM Cell for time t



https://blog.floydhub.com/a-beginners-guide-on-recurrent-neural-networks-with-pytorch/

21

Imperial College London

Detailed Calculations in an LSTM cell

Vanilla RNN

$$h_t = \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

LSTM

"Input" Gate
"Forget" Gate
"Output" Gate

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W' \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

"internal" Cell State: $c_t = f \odot c_{t-1} + i \odot g$

Hidden State: $h_t = o \odot \tanh(c_t)$

Element-wise Multiplication

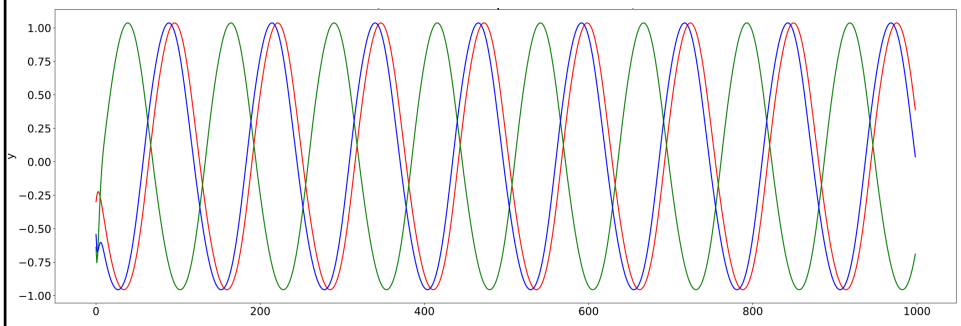
https://www.youtube.com/watch?v=6niqTuYFZLQ

22

Imperial College London

Extrapolating a Sinusoid Using LSTMs (1)

Input (synthetic) Data



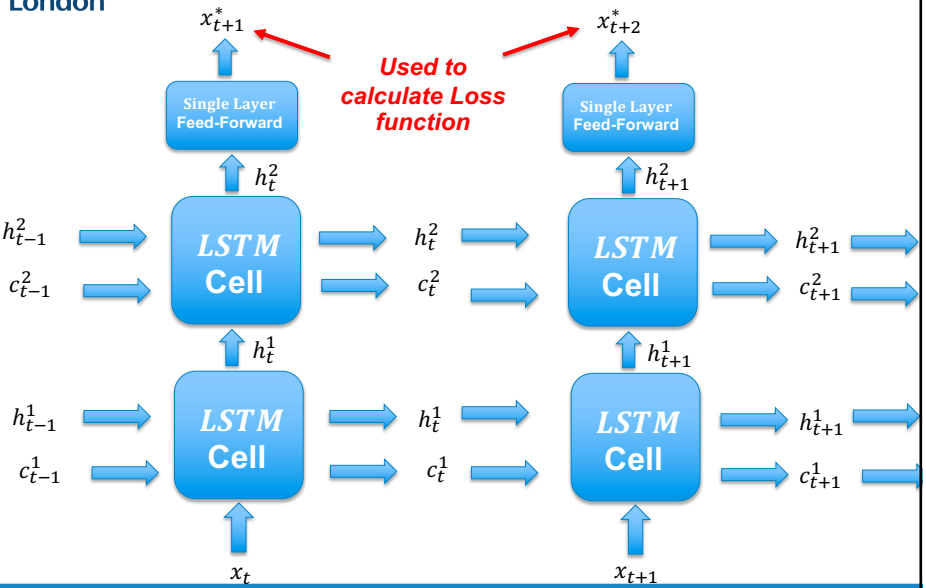
Question : how to interpolate beyond x=1000?

https://github.com/pytorch/examples/tree/master/time_sequence_prediction

23

Imperial College London

Stacked LSTM Cells



https://blog.floydhub.com/a-beginners-guide-on-recurrent-neural-networks-with-pytorch/

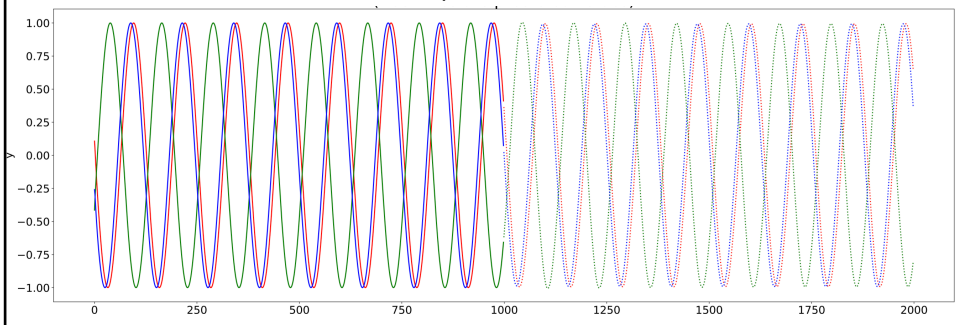
24

Imperial College London

Extrapolating a Sinusoid Using LSTMs

Training Set: 97 sinusoids of 1000 points each

Result of Extrapolation of 3 Test Sinusoids



https://github.com/pytorch/examples/tree/master/time_sequence_prediction

25

Imperial College London

Conclusions on RNNs and LSTMs

- Recurrent Neural Networks (RNNs) the basic approach for sequential data
- RNNs suffer from exploding and vanishing gradients and only preserve short-term memory
- Long-Term Short-Term Memory (LSTM) networks address these two issues

26

Imperial College London

May 11th, 2020

Probabilities for Deep Learning

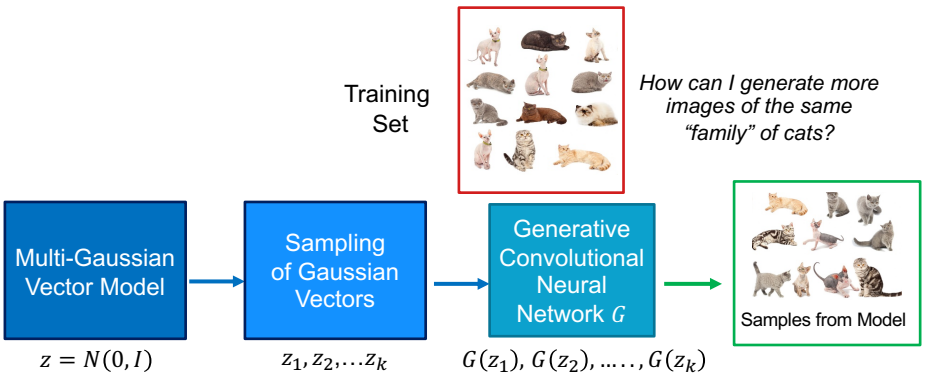
Olivier Dubrule and Navjot Kukreja

27

Imperial College London

Objectives of the Day

- Introduce Basic Probability Concepts Required to Understand Machine Learning, and in particular *Generative Networks*



28

Imperial College London

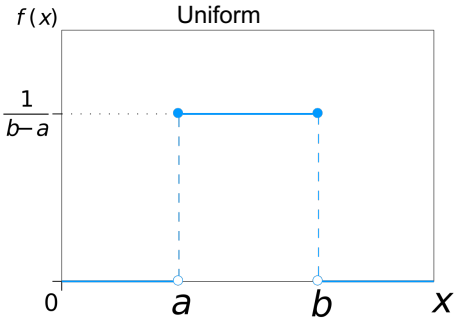
Probabilities for Deep Learning

1. Gaussian, Uniform and Bernouilli pdfs in 1 and n dimensions
2. Maximum Likelihood
3. Comparing Probability Density Functions (PDFs)

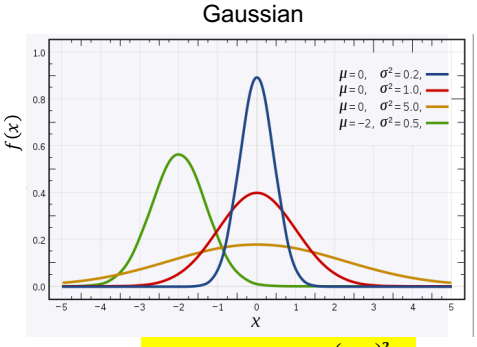
29

Imperial College London

The Uniform and the Normal (or Gaussian) pdfs



Gaussian



$f(x) = \frac{1}{b-a}$ *if* $x \in [a, b]$ and $f(x) = 0$ *if* $x \notin [a, b]$

$f(x)$ also written $\mathrm{U}(x; a, b)$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$f(x)$ also written $N(x; \mu, \sigma^2)$

https://en.wikipedia.org/wiki/Normal_distribution

30

Imperial College London

General Properties of a PDF $f(x)$

$$\mathrm{P}(a < X < b) = \int_a^b f(x)dx \qquad \int_{-\infty}^{+\infty} f(x)dx = 1$$

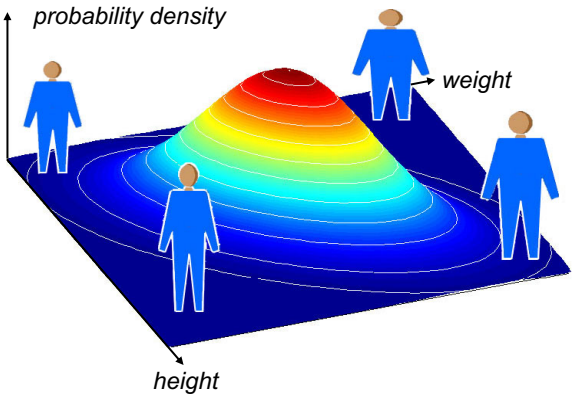
Mean $E(X) = \mu = \int_{-\infty}^{+\infty} xf(x)dx$

Variance
$$Var(X) = \sigma^2 = \int_{-\infty}^{+\infty} (x-\mu)^2 f(x)dx$$
$$= E[(X-\mu)^2] = E(X^2) - \mu^2$$

31

Imperial College London

Dealing with More Than One Dimension



32

Imperial College London

Covariance & Correlation Coefficient of Two Random Variables

If X_1 has mean μ_1 and standard deviation σ_1
If X_2 has mean μ_2 and standard deviation σ_2

$$Cov(X_1, X_2) = E[(X_1 - \mu_1)(X_2 - \mu_2)]$$

The **correlation coefficient** ρ is defined as $\rho = \frac{Cov(X_1, X_2)}{\sigma_1 \sigma_2}$ or $Cov(X_1, X_2) = \rho\, \sigma_1 \sigma_2$

ρ has the following properties:

$$-1 \le \rho \le 1$$

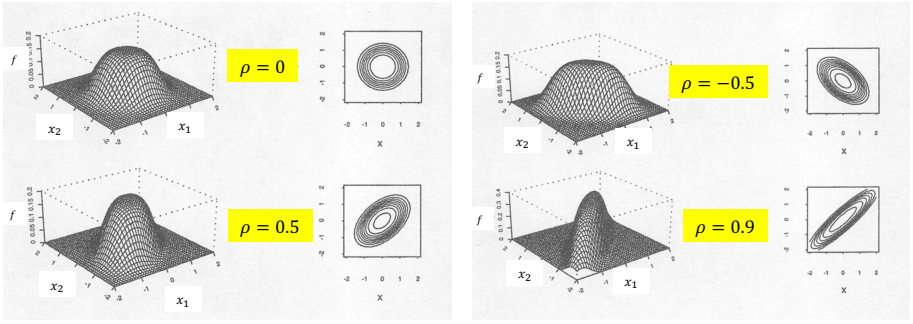
If $Cov(X_1, X_2) = 0$ or $\rho = 0$, then X_1 and X_2 are uncorrelated

If $\rho = -1$ or +1, there is a perfect linear relationship between X_1 and X_2

33

Imperial College London

Examples of Bivariate Normal Distributions



Examples of Bivariate Normal Densities of (x_1, x_2) with $\mu_1 = \mu_2 = 0$, $\sigma_1 = \sigma_2 = 1$

https://www.ilri.org/biometrics/Publication/Full%20Text/Linear_Mixed_Models/AppendixD.htm

34

Imperial College London

Writing the Bivariate Normal Density

If μ is the 2x1 mean vector and Σ is the 2x2 "Variance-Covariance" Matrix of the Bivariate Gaussian : (X_1, X_2)

$$\Sigma = \begin{pmatrix} Var(X_1) & Cov(X_1,X_2) \\ Cov(X_1,X_2) & Var(X_2) \end{pmatrix} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

$$f(x_1, x_2) = \frac{1}{2\pi}\frac{1}{\sqrt{\det(\Sigma)}}\exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]$$

https://www.ilri.org/biometrics/Publication/Full%20Text/Linear_Mixed_Models/AppendixD.htm

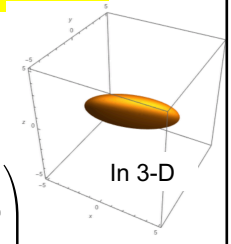
35

Imperial College London

Multivariate Normal pdf of a Random Vector $(X_1, \ldots X_n)$

$$f(x) = \frac{1}{(2\pi)^{n/2}}\frac{1}{\sqrt{\det(\Sigma)}}\exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]$$

$\boldsymbol{x}$ is the $n\times1$ vector $\quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$



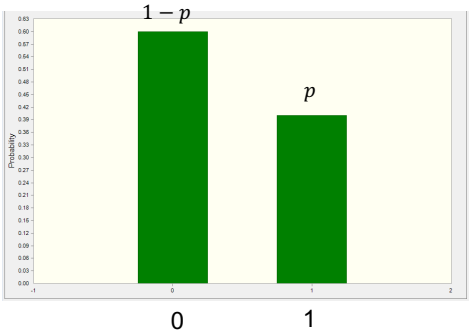
μ is the $n\times1$ expectation vector $\quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix} = \begin{pmatrix} E(X_1) \\ E(X_2) \\ \vdots \\ E(X_n) \end{pmatrix}$

Σ is the $n\times n$ variance-covariance matrix $\quad \Sigma = \left(\left(Cov(X_i, X_j)\right)_{i,j=1,\ldots,n}\right)$

36

Imperial College London

Probability Distribution of a Bernouilli Random Variable



A Bernouilli variable takes the value 1 with probability p and 0 with probability $(1-p)$, and we can write that the probability that it is equal to x is:

$$b(x) = p^x(1-p)^{1-x}$$

37

Imperial College London

Independent and Identically Distributed (IID)

*In probability theory, a sequence or collection of random variables is independent and identically distributed (**i.i.d.** or **iid** or **IID**) if each random variable has the same probability distribution as the others and they all are mutually independent.*

When treating m samples from a training or test dataset (for instance a set of images), it is assumed they are IID.

38

Imperial College London

Probabilities for Deep Learning

1. Gaussian, Uniform and Bernouilli pdfs
2. Maximum Likelihood
3. Comparing Probability Density Functions (PDFs)

39

Imperial College London

Problem Adressed by the Maximum Likelihood Method

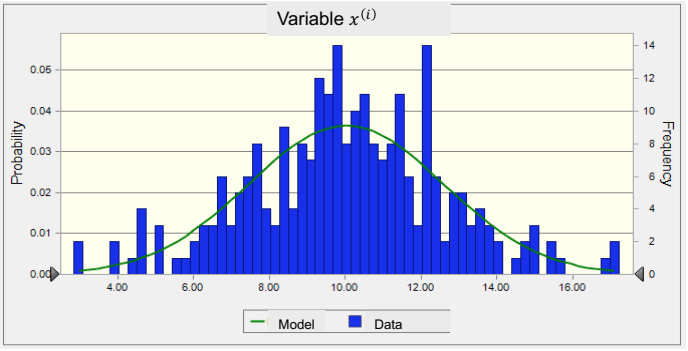
Number of IID Samples

$m = 250$

Number of Features

$n = 1$

How to calculate the normal distribution that best fits these data?



40

Imperial College London

Problem Addressed by the Maximum Likelihood Method

Assume we have m *independent (IID)* data points $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$

Each data point is $x^{(i)}$ is composed of n features.

We want to fit a multivariate (for instance multivariate Gaussian) pdf $p_\theta(x)$ of dimension n to these m samples.

Maximum likelihood consists of calculating the parameters θ such that the m samples maximize their likelihood.

But what is the likelihood? The likelihood is the probability to obtain the m sample values $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ assuming that $p_\theta(x)$ is the probability of x.

41

Imperial College London

Example of Likelihood Calculation for a Normal pdf

Assume we have 4 *independent (IID)* data points $x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}$ with:

$$x^{(1)} = -1,\ x^{(2)} = 0,\ x^{(3)} = 1,\ x^{(4)} = 2$$

Here each data point $x^{(i)}$ is composed of just $n = 1$ feature. We want to fit a Normal distribution $N(x; \mu, \sigma^2)$ to these four data points.

The likelihood of $x^{(1)}$ is: the value of $N(x; \mu, \sigma^2)$ for $x = x^{(1)} = -1$:

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x^{(1)}-\mu}{\sigma}\right)^2} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{-1-\mu}{\sigma}\right)^2}$$

And the likelihood of the IID sequence $(x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$ is:

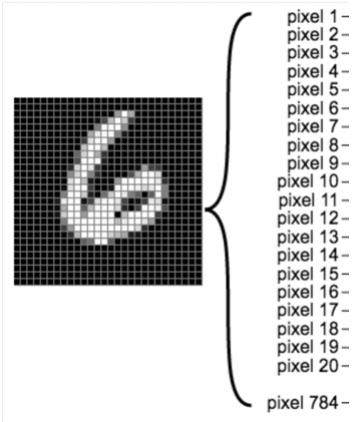
$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{-1-\mu}{\sigma}\right)^2} \times \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{0-\mu}{\sigma}\right)^2} \times \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{1-\mu}{\sigma}\right)^2} \times \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{2-\mu}{\sigma}\right)^2}$$

42

Imperial College London

One Sample can also be an Image , as in MNIST

$n = 784$ (features)



43

Imperial College London

Likelihood and Maximum Likelihood for a Dataset of Images

The pdf $p_\theta(x_1, x_2, \ldots, x_n)$ is parametrized by θ. If there is just one image $x^{(1)} = \left(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}\right)$ in the dataset, its likelihood is defined as:

$$Likelihood\ of\ image\ x^{(1)} = p_\theta\left(x^{(1)}\right) = p_\theta\left(x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)}\right)$$

The Maximum Likelihood estimate θ_{ML} of θ is calculated as

$$\theta_{ML} = argmax\left(p_\theta\left(x^{(1)}\right)\right) \quad \text{or} \quad \theta_{ML} = argmax\left(\log p_\theta\left(x^{(1)}\right)\right)$$

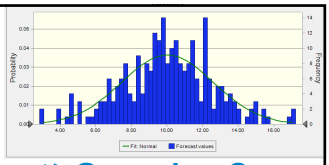
If there are m *IID* images $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ in the dataset

$$\theta_{ML} = argmax\left(p_\theta\left(x^{(1)}\right)p_\theta\left(x^{(2)}\right)\ldots p_\theta\left(x^{(m)}\right)\right)$$

$$= argmax\left(\log\left(p_\theta\left(x^{(1)}\right)p_\theta\left(x^{(2)}\right)\ldots p_\theta\left(x^{(m)}\right)\right)\right) = \theta_{ML} = argmax\left(\sum_{i=1}^{m} \log p_\theta\left(x^{(i)}\right)\right)$$

44

Imperial College London



Maximum Likelihood in the Univariate $(n = 1)$ Gaussian Case

Goal : fit a Gaussian pdf to a dataset

$$N(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\, exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

Here the θ parameters are μ and σ

We have $p_\theta(x) = N(x;\mu,\sigma^2)$

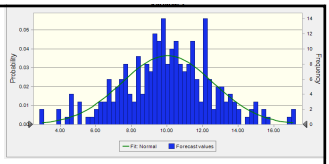
Hence the log-Likelihood $\sum_{i=1}^{m}\left(\log p_\theta\left(x^{(i)}\right)\right) = -\frac{1}{2}\sum_{i=1}^{m}\left(\frac{x^{(i)}-\mu}{\sigma}\right)^2 - m\log\sigma - \frac{m}{2}\log 2\pi$

The Maximum Likelihood Estimate for a Gaussian leads to the L2 norm!

https://stats.stackexchange.com/questions/351549/maximum-likelihood-estimators-multivariate-gaussian
https://onlinecourses.science.psu.edu/stat414/node/191/

45

Imperial College London



Exercise

We have m IID values of real numbers $(x_i)_{i=1\ldots m}$

We want to calculate the parameters of a normal distribution $N(x;\mu,\sigma^2)$ that best fits these *m* values.

What is the maximum likelihood estimate μ_{ML} of μ?

What is the maximum likelihood estimate σ^2_{ML} of σ^2?

$$\mu_{ML} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} \qquad \sigma^2_{ML} = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)}-\mu\right)^2$$

46

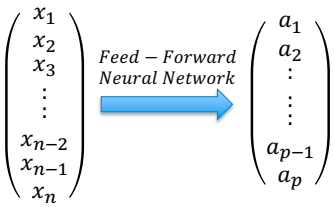
Imperial College London

A Simple Way to See Supervised Neural Networks

Suppose we have m pairs of data.
Each pair is composed of a vector of dimension n and a vector of dimension p (the labels).

A neural network is simply a function that maps any vector of dimension n into a (discrete or continuous) vector of dimension p.

In order to calculate the parameters of this function, we train the parameters of the neural network by back-propagation using the m pairs of data as Training Set.



Input Vector

Output Labels

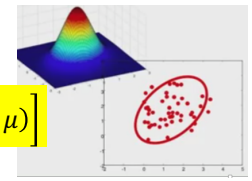
47

Imperial College London

Can be skipped

Maximum Likelihood in the Multivariate Gaussian Case (1)

Idea : fit a multivariate Gaussian to a n-dimension dataset



$$f(x) = (2\pi)^{-\frac{n}{2}}(det\Sigma)^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right]$$

Here the θ parameters are the vector μ and the matrix Σ

With the previous notation we have $p_\theta(x) = N(x;\mu,\Sigma)$

Hence

$$\sum_{i=1}^{m}\left(\log p_\theta\left(x^{(i)}\right)\right) = -\frac{nm}{2}\log 2\pi - \frac{m}{2}\log(det\Sigma) - \frac{1}{2}\sum_{i=1}^{m}\left(x^{(i)}-\mu\right)^T\Sigma^{-1}\left(x^{(i)}-\mu\right)$$

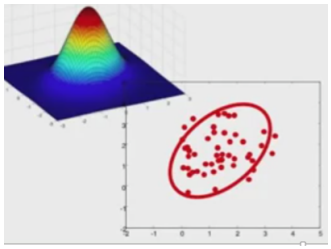
https://stats.stackexchange.com/questions/351549/maximum-likelihood-estimators-multivariate-gaussian

48

Imperial College London

Can be skipped

Maximum Likelihood in the Multivariate Gaussian Case (2)



Maximum Likelihood Estimates

$$\mu_{ML} = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

$$\Sigma_{ML} = \frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)} - \mu_{ML}\right)^T \left(x^{(i)} - \mu_{ML}\right)$$

https://stats.stackexchange.com/questions/351549/maximum-likelihood-estimators-multivariate-gaussian

49

Imperial College London

Maximum Likelihood for a Bernouilli Distribution

A Bernouilli distribution has just one parameter p

Suppose we have just one sample x_1 (which has the value 1 or 0)

Its likelihood is: $b(x_1) = p^{x_1}(1-p)^{1-x_1}$

Its log-likelihood is: $\log b(x_1) = x_1 \log p + (1 - x_1)\log(1-p)$

If we have m samples x_i, their log-likelihood is:

$\sum_{i=1\ldots m}(x_i \log p + (1 - x_i)\log(1-p))$ **(this is equal to minus the cross-entropy!)**

The maximization leads, unsurprisingly, to : $p_{ML} = \frac{1}{m}\sum_{i=1\ldots m} x_i$.
(p_{ML} is the proportion of samples equal to 1)

50

Imperial College London

Probabilities for Deep Learning

1. **Gaussian , Uniform and Bernouilli pdfs**
2. **Maximum Likelihood**
3. **Comparing Probability Density Functions (PDFs)**

51

Imperial College London

Compare two pdfs: the Kullback-Leibler (KL) Divergence

For two pdfs $p(x)$ and $q(x)$:

$$D_{KL}(p\|q) = \int_{-\infty}^{+\infty} p(x)\log p(x)\,dx - \int_{-\infty}^{+\infty} p(x)\log q(x)\,dx$$

Two Fundamental Properties

- $D_{KL}(p\|q)$ is positive, and 0 if $p(x)$ and $q(x)$ identical
- Asymmetry: $D_{KL}(p\|q) \neq D_{KL}(q\|p)$

52

Imperial College London

KL Divergence between $f = N(x; \mu_1, \sigma_1^2)$ and $g = N(x; \mu_2, \sigma_2^2)$

$$KL(f, g) = -\frac{1}{2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} + \log \frac{\sigma_2}{\sigma_1}$$

(see Exercise 3 for calculation of KL Divergence for Gaussians and Multi-Gaussians)

From Deep Learning, by Goodfellow et al, 2016

53

Imperial College London

KL Divergence versus Maximum Likelihood

If the Training Set consists of m *IID* data points $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$, the KL Divergence $D_{KL}(p_d \| p_\theta)$ between the experimental distribution p_d of the Training Set and any theoretical distribution p_θ is:

$$D_{KL}(p_d \| p_\theta) = \int_{-\infty}^{+\infty} p_d(x) \log p_d(x)\, dx - \int_{-\infty}^{+\infty} p_d(x) \log p_\theta(x)\, dx$$

Minimizing $D_{KL}(p_d \| p_\theta)$ in the parameters θ is equivalent to maximizing the second term:

$$\int_{-\infty}^{+\infty} p_d(x) \log p_\theta(x)\, dx$$

But this is the expression of the expectation of $\log p_\theta(x)$ calculated over the Training Set! Hence

$$\theta = argmax\left(\frac{1}{m}\sum_{i=1}^{m} \log p_\theta(x^{(i)})\right)$$

Minimizing the KL Divergence is equivalent to maximizing the Likelihood!

54

Imperial College London

Can be skipped

Other Criteria for Selecting the "Best" Model (1)

There are three main types of criteria:

1. Criteria based on the performance of the model on the validation and test sets. But this approach requires a lot of data.
2. Resampling techniques achieve the same as above but with a small dataset. An example is k-fold cross-validation .

The above methods only model performance, regardless of model complexity.

A third model selection approach attempts to combine the complexity of the model with its performance into a single loss function, then select the model that minimizes this loss.

https://machinelearningmastery.com/probabilistic-model-selection-measures/

55

Imperial College London

Can be skipped

Other (Positive) Criteria for Selecting the "Best" Model (2)

m is the number of data in Training Set, p is number of model parameters

1. Minimize Akaike Information Criteria (AIC)

$$\text{AIC} = -\frac{2}{m} Log\ Likelihood + \frac{2}{m} p$$

2. Minimize Bayesian Information Criterion (BIC)

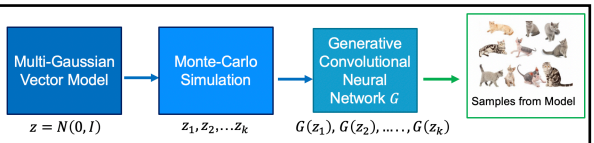
$$\text{BIC} = -2\ Log\ Likelihood + Log(m)\ p$$

BIC penalizes complex models more than AIC, which means that BIC prefers more "parsimonious" models

https://machinelearningmastery.com/probabilistic-model-selection-measures/

56

Imperial College London



Conclusions on Probabilities

- The basic pdfs used in Deep Learning are Bernouilli for discrete variables and (Multivariate) Gaussian or Uniform for continuous variables.
- Maximum Likelihood is a key approach in Deep Learning and applying it to Bernouilli and (Multivariate) Gaussian random variables leads respectively to the minimization of the cross-entropy for classification or the L2 norm for regression.
- The Kullback-Leibner (KL) Divergence is used to calculate the dissimilarity between two pdfs. Minimizing it allows one pdf to be adjusted to another.