UNIVERSITY
OF OSLO

IN4310 An Introduction to Deep Learning Theory

Tobias Müller

tobiasmu@ifi.uio.no



Some Questions

What are we learning in a neural network?

Why is that cross-entropy loss defined so weirdly?

How good can our neural network be in the absolute best case? Is there a bound?

How do intermediate layers transform the information from the input to the output?

Do neural networks forget something?

Uncertainty in a Die

Let us consider two dice. A regular one, and a skewed one. They have the following probabilities

Die	1	2	3	4	5	6
A	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
B	$\frac{1}{2}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$

Which die is more uncertain?

Uncertainty in a Die

Let us consider two dice. A regular one, and a skewed one. They have the following probabilities

Die	1	2	3	4	5	6
A	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$
B	$\frac{1}{2}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$

Which die is more uncertain?

Where do you expect more bits to communicate the result?

Brings us to ENTROPY

Entropy

We want to measure the uncertainty in a random variable X (only discrete for now).
The surprise of some outcome x is $-\log(p(X = x))$.
Entropy is the expected surprise, given the distribution of X.

$$H(X) = -\sum_x p(X = x) \log(p(X = x)) = E(-\log(p(X)))$$

Intuitively, high entropy means there is much uncertainty in the random variable, low entropy means the variable is mostly predetermined.

1. Die X with numbers 1 to n:

$$H(X) = -\sum_{i=1}^{n} p_i \log(p_i) = -\sum_{i=1}^{n} \frac{1}{n} \log(\frac{1}{n}) = -\log(\frac{1}{n}) = \log(n)$$

1. Die X with numbers 1 to n:

$$H(X) = -\sum_{i=1}^{n} p_i \log(p_i) = -\sum_{i=1}^{n} \frac{1}{n} \log(\frac{1}{n}) = -\log(\frac{1}{n}) = \log(n)$$

´ 2. A weird coin that lands on the edge with probability $1/2$

H	T	E
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$

1. Die X with numbers 1 to n:

$$H(X) = -\sum_{i=1}^{n} p_i \log(p_i) = -\sum_{i=1}^{n} \frac{1}{n} \log(\frac{1}{n}) = -\log(\frac{1}{n}) = \log(n)$$

´ 2. A weird coin that lands on the edge with probability $1/2$

H	T	E
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$

$$H(X) = -\left(\frac{1}{2}\log\frac{1}{2} + 2 \cdot \frac{1}{4}\log\frac{1}{4}\right) = -(-\frac{1}{2} - 1) = \frac{3}{2} < \log(3)$$

Other Types of Entropy

We are interested in more than one random variable. Say X and Y. Then there is

- The joint entropy $H(X, Y)$

$$H(X, Y) = -\sum_{x,y} p(x, y) \log(p(x, y)) = E(-\log(p(X, Y)))$$

"How much uncertainty is in X and Y together"

- The conditional entropy $H(Y|X)$

$$H(Y|X) = \sum_{x} p(x) H(Y|X = x) = -\sum_{x} p(x) \sum_{x,y} p(y|x) \log p(y|x)$$
$$= E(-\log p(Y|X))$$

"How much uncertainty is left in Y after knowing X"

Properties of Entropy

Entropy has some nice properties.

- $H(0) = ?$
- $H(X, Y) = ?$ (if X and Y are independent)

Properties of Entropy

Entropy has some nice properties.

- $H(0) = ?$
- $H(X, Y) = ?$ (if X and Y are independent)

$H(0) = 0$

Properties of Entropy

Entropy has some nice properties.

- $H(0) = ?$
- $H(X, Y) = ?$ (if X and Y are independent)

$H(0) = 0$

For the second, we compute

$$\begin{aligned} H(X,Y) &= \sum_{x,y} p(x)p(y)\log(p(x)p(y)) = \sum_{x,y} p(x)p(y)(\log(p(x)) + \log(p(y))) \\ &= \sum_{x,y} p(x)p(y)\log(p(x)) + \sum_{x,y} p(x)p(y)\log(p(y)) \\ &= H(X) + H(Y) \end{aligned}$$

Second means that entropy is additive for independent variables.

Kullback-Leibler (KL) Divergence

Let us now think only about probability distributions on the same domain. We want to know how similar they are. Let

- P be the true distribution
- Q be our model.

Kullback-Leibler (KL) Divergence

Let us now think only about probability distributions on the same domain. We want to know how similar they are. Let

- P be the true distribution
- Q be our model.

How close is Q to the truth P?
Say (information theoretically), we encode data from P with Q. How much additional uncertainty do we expect?

Kullback-Leibler (KL) Divergence

Let us now think only about probability distributions on the same domain. We want to know how similar they are. Let

- P be the true distribution
- Q be our model.

How close is Q to the truth P?
Say (information theoretically), we encode data from P with Q. How much additional uncertainty do we expect?

The cross-entropy $H(P;Q)$

$$H(P;Q) = -\sum_x P(x)\log(Q(x))$$

Kullback-Leibler (KL) Divergence

Let us now think only about probability distributions on the same domain. We want to know how similar they are. Let

- P be the true distribution
- Q be our model.

How close is Q to the truth P?
Say (information theoretically), we encode data from P with Q. How much additional uncertainty do we expect?

The cross-entropy $H(P;Q)$

$$H(P;Q) = -\sum_x P(x)\log(Q(x))$$

Also we have the entropy of P

$$H(P) = -\sum_x P(x)\log(P(x))$$

KL Divergence

How to get the additional uncertainty?

KL Divergence

How to get the additional uncertainty? By computing the difference.

$$D_{KL}(P\|Q) = H(P;Q) - H(P)$$

KL Divergence

How to get the additional uncertainty? By computing the difference.

$$D_{KL}(P\|Q) = H(P;Q) - H(P)$$

This can be rewritten as

$$D_{KL}(P\|Q) = -\sum_x P(x)\log(Q(x)) + \sum_x P(x)\log(P(x))$$

$$= \sum_x P(x)(\log(P(x)) - \log(Q(x))) = \sum_x P(x)(\log(\frac{P(x)}{Q(x)}))$$

KL Divergence

Standard definition is

$$D_{KL}(P\|Q) = \sum_x P(x) \log(\frac{P(x)}{Q(x)})$$

KL Divergence

Standard definition is

$$D_{KL}(P\|Q) = \sum_x P(x) \log(\frac{P(x)}{Q(x)})$$

Important properties

- $D(P\|Q) = 0$ if and only if $P = Q$
- In general $D(P\|Q) \neq D(Q\|P)$, in particular D_{KL} is not a metric.

Relationship between Random Variables

Throw two dices. Let us think about the following random variables

- X: result of first die
- Y: result of second die
- $Z = X + Y$

I tell you that X is 5.
What do you know about Y? about Z?
Nothing about Y. But something about Z. (e.g., it will not be 4)

Mutual Information

How to quantify how much knowing one random variable tells us about another?

Idea: **Reduction of uncertainty** Formally:

$$I(X;Y) = H(X) - H(X|Y)$$

This is the mutual information of X and Y.

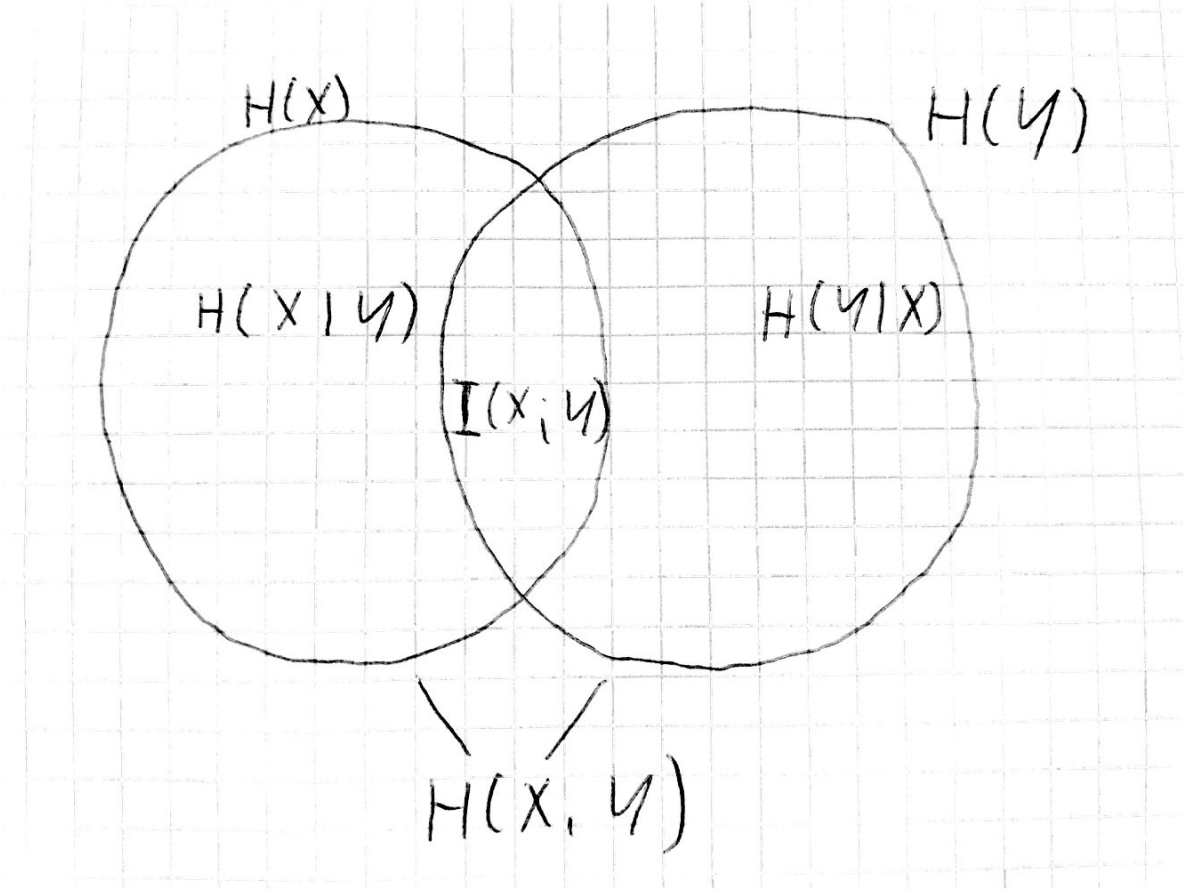
Equivalently, with $p(X,Y)$ the joint distribution of X and Y

$$I(X;Y) = D_{KL}(p(X,Y)\|p(X)p(Y)) = \sum_{x,y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right)$$

Key property: $I(X;Y) = 0$ if and only if X and Y are independent.

Intuitive Overview

The relationship between all introduced quantities can be described by a venn diagram



Standard supervised learning setting: Random variables X, Y with joint distribution $p(X, Y)$.

Want to learn $f^* : X \to Y$, such that $f^* = \text{argmin}_{f \in \mathcal{F}} E(\ell(Y, f(X)))$

Let us introduce a second view on this: If we knew

$$P(Y|X),$$

then

$$f^*(x) = \text{argmax}_{k \in \mathcal{Y}} P(Y = k|X = x)$$

solves classification with 0-1 loss.

Cross Entropy Loss Function

In classification: Interpret softmax output as distribution, gives us distribution $q(Y|X)$ with

$$q(Y = k|X = x) = f(x)_k$$

Cross Entropy Loss Function

In classification: Interpret softmax output as distribution, gives us distribution $q(Y|X)$ with

$$q(Y = k|X = x) = f(x)_k$$

Then, we compute the KL divergence with the true conditional distribution

$$D_{KL}(P(Y|X)\|q(Y|X)) = H(P(Y|X), q(Y|X)) - H(P(Y|X))$$

Cross Entropy Loss Function

In classification: Interpret softmax output as distribution, gives us distribution $q(Y|X)$ with

$$q(Y = k|X = x) = f(x)_k$$

Then, we compute the KL divergence with the true conditional distribution

$$D_{KL}(P(Y|X)\|q(Y|X)) = H(P(Y|X), q(Y|X)) - H(P(Y|X))$$

We have

$$\begin{aligned} H(P(Y|X), q(Y|X)) &= -E_{x\sim P(X)} \sum_y P(y|x) \log(q(y|x)) \\ &= E_{(x,y)\sim P(X,Y)} - \log(q(y|x)) \end{aligned}$$

Cross Entropy Loss Function

Therefore

$$H(P(Y|X), q(Y|X)) = E_{(x,y)\sim P(X,Y)} - \log(q(y|x))$$

is approximated by the cross entropy loss

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{i=1}^{N} \log q(y_i \mid x_i)$$

Information Bottleneck in Neural Networks

We want to understand and analyze what happens to the information inside a neural network.

Information Bottleneck in Neural Networks

We want to understand and analyze what happens to the information inside a neural network.

Write down the layers

$$X \to X_1 \to X_2 \to \ldots \to X_n = \hat{Y}$$

Information Bottleneck in Neural Networks

We want to understand and analyze what happens to the information inside a neural network.

Write down the layers

$$X \to X_1 \to X_2 \to \ldots \to X_n = \hat{Y}$$

Question: How good is the representation of some layer T (one of the $(X_1, \ldots, X_n)$

Information Bottleneck in Neural Networks

We want to understand and analyze what happens to the information inside a neural network.

Write down the layers

$$X \to X_1 \to X_2 \to \ldots \to X_n = \hat{Y}$$

Question: How good is the representation of some layer T (one of the $(X_1, \ldots, X_n)$

Idea (Tishby 2015): Interpret it as a "bottleneck", measure $I(X;T)$ and $I(T;Y)$.

Want: High $I(T;Y)$, low $I(X;T)$

Information Bottleneck in Neural Networks

We want to understand and analyze what happens to the information inside a neural network.

Write down the layers

$$X \to X_1 \to X_2 \to \ldots \to X_n = \hat{Y}$$

Question: How good is the representation of some layer T (one of the $(X_1, \ldots, X_n)$

Idea (Tishby 2015): Interpret it as a "bottleneck", measure $I(X;T)$ and $I(T;Y)$.

Want: High $I(T;Y)$, low $I(X;T)$
That is (very informally) the Information Bottleneck Problem

Information Bottleneck Problem

Having a high $I(T;Y)$ makes sense, because we get the following bound. We assume that our model q only depends on X via T (as in a neural network)

$$\begin{aligned}
E(-\log q(Y|X)) &= E(-\log q(Y|T)) \\
&= E(-\log P(Y|T)) + E(\log\left(\frac{P(Y|T)}{q(Y|T)}\right)) \\
&= H(Y|T) + D_{KL}(P(Y|T)\|q(Y|T)) \\
&\geq H(Y|T) = H(Y) - (H(Y) - H(Y|T)) \\
&= H(Y) - I(Y;T)
\end{aligned}$$

So $I(Y;T)$ naturally bounds the loss value.

What about compression?

So far, we only looked at $I(T;Y)$. What about $I(X;T)$?

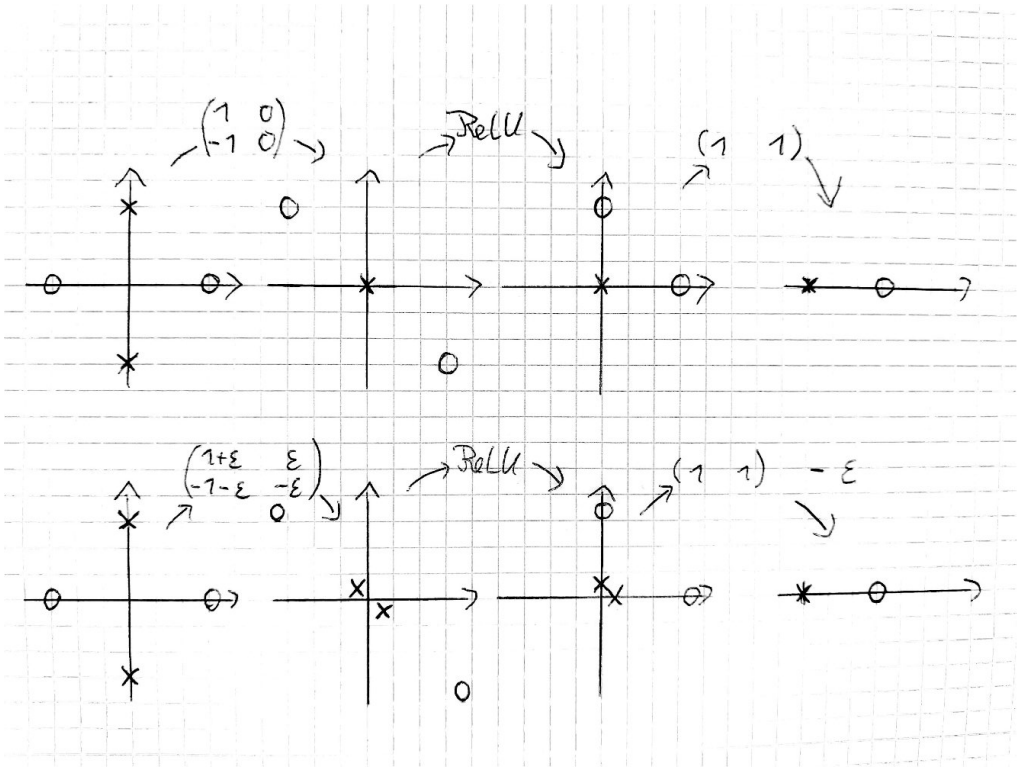
Measures "compression" of the input.

There is an issue here

- $I(X;T)$ is, formally seen, infinity for continuous distributions and T being a function of X
- So, we need some approximation (binning, adding noise). Hard to interpret

Simple Example

XOR Problem: Classify the set $\{(1,0),(-1,0)\}$ as 0 and $\{(0,1),(0,-1)\}$ as 1. Both networks sketched here solve the problem.



Consider the second random variable, call it T. In both cases $I(T;Y) = I(X;Y) = 1\text{bit}$. But in the first case $I(X;T) = 3/2\text{bit}$, in the second $I(X;T) = 2\text{bit}$

The Information Bottleneck Discussion

This information bottleneck idea applied to neural networks was proposed by Shwartz and Tishby (2017). They find the following curves

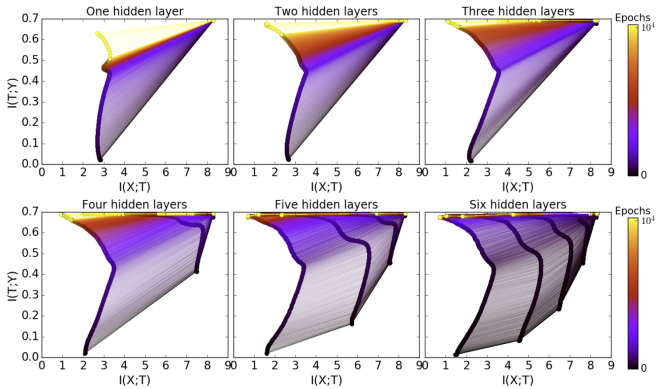


Figure 5: **The layers information paths during the SGD optimization for different architectures.** Each panel is the *information plane* for a network with a different number of hidden layers. The width of the hidden layers start with 12, and each additional layer has 2 fewer neurons. The final layer with 2 neurons is shown in all panels. The line colors correspond to the number of training epochs.

The observe: First "fitting", here both mutual informations increase. Then "compression" where $I(X;T)$ shrinks.

Debate: Later work claims this is an artifact of tanh activation and binning.