An Information Theoretic View on Learning of Artificial Neural Networks

Emilio Balda, Arash Behboodi, Rudolf Mathar

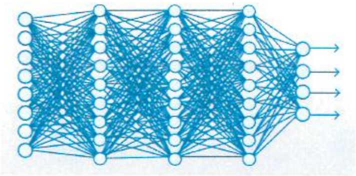
Institute for Theoretical Information Technology
Faculty of Electrical Engineering and Information Technology

Chair for Theoretical Information Technology | RWTH AACHEN UNIVERSITY

Some Questions

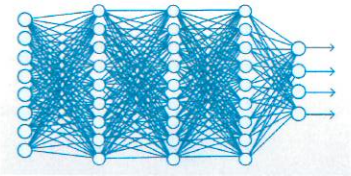
- The brain seems to be a biological neural network. Is its functionality really understood?

Some Questions



- The brain seems to be a biological neural network. Is its functionality really understood?
- Artificial neural networks, used in deep learning, show great success. Why exactly?

Some Questions



- The brain seems to be a biological neural network. Is its functionality really understood?
- Artificial neural networks, used in deep learning, show great success. Why exactly?
- Is there a way to explain how BioNNs and ANNs learn?
- Are (artificial) neural networks trained in a way to maximize information flow?
- Are there analogies between ANNs and BioNNs?

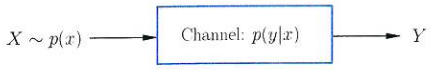
Overview

- Basic concepts from information theory
- Biological neural networks (BioNN)
- Artificial neural networks (ANN)
- An information theoretic approach
- Noisy training data
- Outlook

Overview

- Basic concepts from information theory
- Biological neural networks (BioNN)
- Artificial neural networks (ANN)
- An information theoretic approach
- Noisy training data
- Outlook

Channels



How much information can you get across a noisy channel?

$$H(X) = -\sum_i p(x_i) \log p(x_i)$$ **entropy** of X

$$H(Y \mid X) = -\sum_{i,j} p(x_i, y_j) \log p(x_i|y_j)$$ **conditional entropy** of X given Y

$$I(X;Y) = H(Y) - H(Y \mid X) = H(X) - H(X \mid Y)$$ **mutual information** between X and Y

$$C = \max_{p(x)} I(X;Y)$$ **capacity** of the channel

Interpretation

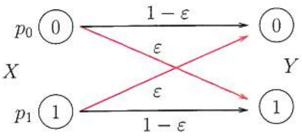
$H(X)$ = Entropy
= uncertainty about the outcome of X

$H(Y \mid X)$ = Conditional entropy
= uncertainty about Y when X is given

$I(X;Y)$ = Mutual information
= reduction in uncertainty about X when Y is given
= amount of information about X provided by Y.

C = Capacity
= maximum amount of information about X provided by Y over all input distributions $p(X)$

Example: Binary Symmetric Channel (BSC)



Mutual Information:

$$I(X;Y) = H\big(p_0(1-\varepsilon) + p_1\varepsilon, \varepsilon p_0 + (1-\varepsilon)p_1\big) - H(\varepsilon, 1-\varepsilon)$$

The maximum of $I(X;Y)$ over all input distributions (p_0, p_1) is attained at the uniform distribution $(p_0^*, p_1^*) = (0.5, 0.5)$.

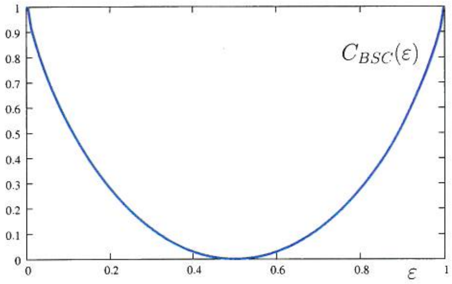
Capacity:

$$C = 1 + (1-\varepsilon)\log_2(1-\varepsilon) + \varepsilon \log_2 \varepsilon$$

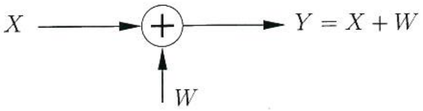
Example: Binary Symmetric Channel (BSC)

Capacity as a function of the error probability ε:

$$C_{BSC}(\varepsilon) = 1 + (1-\varepsilon)\log_2(1-\varepsilon) + \varepsilon \log_2 \varepsilon$$



Example: Gaussian Channel (AWGN)



Average power constraint: $E[X^2] \le M$

Noise distribution is zero-mean Gaussian:

$$W \sim f_W(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\big(-x^2/2\sigma^2\big), \; x \in \mathbb{R}$$

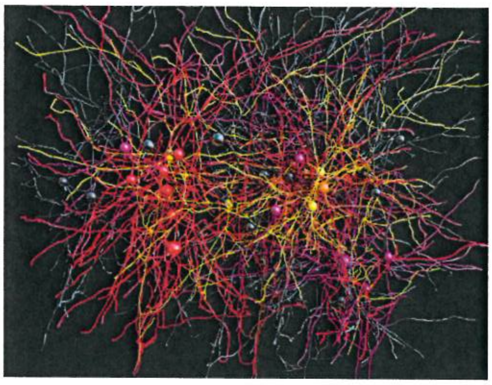
Capacity:

$$C = \frac{1}{2} \ln\big(1 + M/\sigma^2\big) = \frac{1}{2} \ln(1 + SNR)$$

Overview

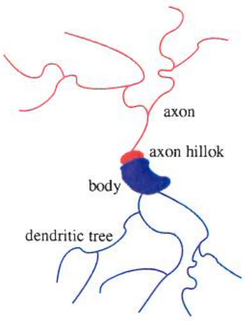
- Basic concepts from information theory
- Biological neural networks (BioNN)
- Artificial neural networks (ANN)
- An information theoretic approach
- [illegible]
- [illegible]

Cortical Neural Networks



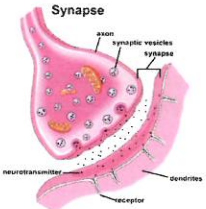
Neuroscientists have constructed a network map of connections between cortical neurons, traced from a 100 terabytes 3D data set. The data were created by an electron microscope in nanoscopic detail, allowing every one of the "wires" to be seen, along with their connections. Some of the neurons are color-coded according to their activity patterns in the living brain. (credit: Clay Reid, Allen Institute; Wei-Chung Lee, Harvard Medical School; Sam Ingersoll, graphic artist)

A Typical Cortical Neuron



- The axon branches to contact other neurons.
- A dendritic tree collects input from other neurons.
- Axons contact dendritic trees at **synapses** and inject spikes of activity.
- An axon hillock generates outgoing spikes whenever enough charge has flowed in at **synapses** to depolarize the cell membrane.

Synapses



- When a spike travels along an axon and arrives at a synapse, vesicles of a transmitter chemical are released.
- The transmitter molecules diffuse through the synaptic cleft and bind to receptor molecules in the membrane of the post-synaptic neuron.
- This opens up holes that allow specific ions to cross.

Synapses

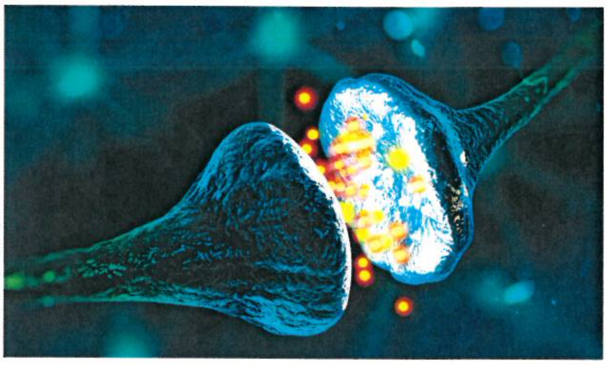
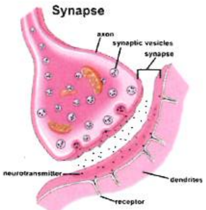


Image Credits: Inside SCIENCE, Andrii Vodolazhskyi

Synapses



- The effectiveness of the synapse can be changed by
  - varying the number of vesicles of the transmitter
  - varying the number of receptor molecules
- Synapses are slow, but
  - they are very small and very low-power
  - they adapt using locally available signals

How (most people think) the brain works

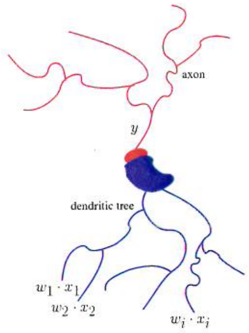
- Each neuron receives input from other neurons
  - A few neurons are connected to receptors.
  - Neurons use spikes to communicate.
- The effect of each input line on the neuron is controlled by synaptic weights
  - Weights can be positive or negative.
- The synaptic weights **adapt** so that the whole network learns to perform useful computations
  - Recognizing objects, understanding language, making plans, controlling the body
- Humans have about 10^{11} neurons each with about 10^4 weights
  - Computations in parallel in a short time, huge bandwidth

Overview

- ▶ Basic concepts from information theory
- ▶ Biological neural networks (BioNN)
- ▶ Artificial neural networks (ANN)
- ▶ An information theoretic approach
- ▶ Noisy training data
- ▶ Outlook

Idealizing Neurons

Substitute spikes by real values x_i, model intensities by weights w_i.

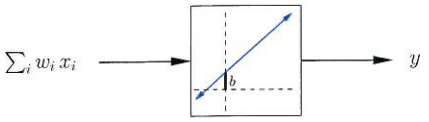


$$y = Q\Big(\sum_i w_i \cdot x_i\Big)$$

What types of **activation functions** Q are appropriate?

Linear Neurons

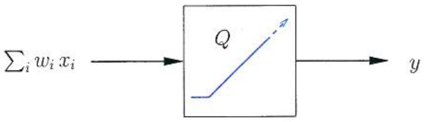
- ▶ First compute a weighted sum of the inputs.
- ▶ Send out a linear transformation of the input.



$$y = Q\Big(\sum_i w_i x_i\Big), \qquad Q(z) = az + b$$

Rectified Linear Neurons

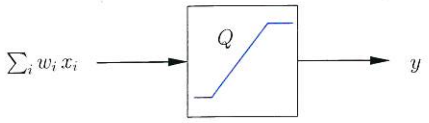
- ▶ First compute a weighted sum of the inputs.
- ▶ Send out a rectified linear function of the weighted sum.



$$y = Q\Big(\sum_i w_i x_i\Big), \qquad Q(z) = \begin{cases} 0, & \text{if } z < 0 \\ z, & \text{if } z \geq 0 \end{cases}$$

Censoring Neurons

- ▶ First compute a weighted sum of the inputs.
- ▶ Send out a censored linear function of the weighted sum.

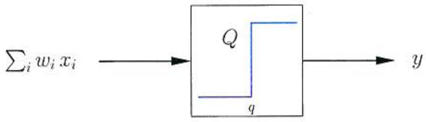


$$y = Q\Big(\sum_i w_i x_i\Big), \qquad Q(z) = \begin{cases} 0, & \text{if } z < 0 \\ z, & \text{if } 0 \leq z < 1 \\ 1, & \text{if } z \geq 1 \end{cases}$$

Binary Threshold Neurons

McCulloch-Pitts (1943) (influenced Von Neumann)

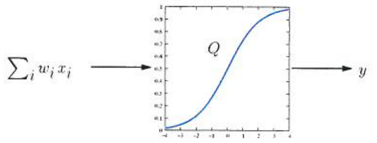
- ▶ First compute a weighted sum of the inputs.
- ▶ Send out a fixed size spike of activity if the weighted sum exceeds a threshold q.



$$y = Q\Big(\sum_i w_i x_i\Big), \qquad Q(z) = \begin{cases} 0, & \text{if } z < q \\ 1, & \text{if } z \geq q \end{cases}$$

Sigmoid Neurons

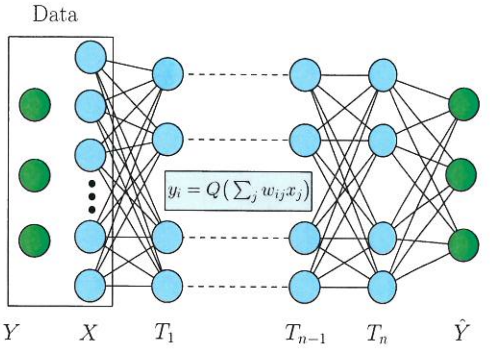
- First compute a weighted sum of the inputs.
- Send out a sigmoid function of the weighted sum.



$$y = Q\Big(\sum_i w_i x_i\Big), \qquad Q(z) = \frac{1}{1+e^{-z}},\ z \in \mathbb{R}$$

- The logistic function with convenient derivatives

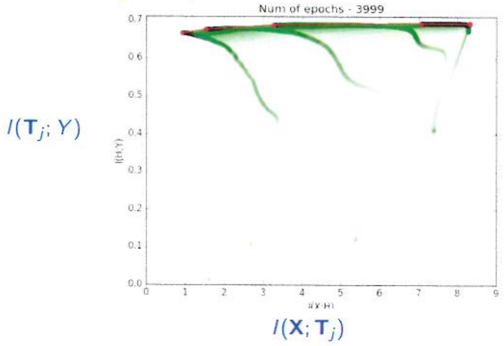
A Feed Forward Classification ANN



Information is passed from input X to output $\hat{Y}$ layer by layer.
The network forms a sequence of consecutive channels:

$Y, \mathbf{X}, \mathbf{T}_1, \dots, \mathbf{T}_{n-1}, \mathbf{T}_n, \hat{Y}$ (a Markov chain)

Naftali Tishby's Experiments



When optimizing parameters of the DNN
- $I(\mathbf{X}; \mathbf{T}_j)$ first increases, then decreases,
- $I(\mathbf{T}_j; Y)$ tends to its max with the number of iterations.

Modeling ANNs

An ANN is specified by weights $\mathbf{W}_\ell$ and biases $\mathbf{b}_\ell$ and the recursion

$$\mathbf{T}_\ell = Q(\mathbf{W}_\ell \mathbf{T}_{\ell-1} + \mathbf{b}_\ell), \quad \ell = 1, \dots, n.$$

Once $\vartheta = (\mathbf{W}_\ell, \mathbf{b}_\ell)_{\ell=1,\dots,n}$ is fixed an ANN is described by a function

$$\hat{\mathbf{y}} = g_\vartheta(\mathbf{x})$$

Input $\mathbf{X}$ to an ANN is modeled as a random variable $(\mathbf{X}, Y)$,

$$\mathbf{X} \in \mathbb{R}^p \text{ with class label } Y \in \{0, 1, \dots, K-1\}$$

$Y = c(\mathbf{X})$ may be a function of $\mathbf{X}$ or noisy with additional random effects.

Modeling ANNs

An ANN decides about the class label of input $\mathbf{X}$ as

$$\hat{Y} = g_\vartheta(\mathbf{X})$$

Of course it may happen that $\hat{Y} \neq Y$.

Expected error = test error

$$\varepsilon = R(g_\vartheta) = P(\hat{Y} \neq Y)$$

Training set are independent samples of $(\mathbf{X}, Y)$

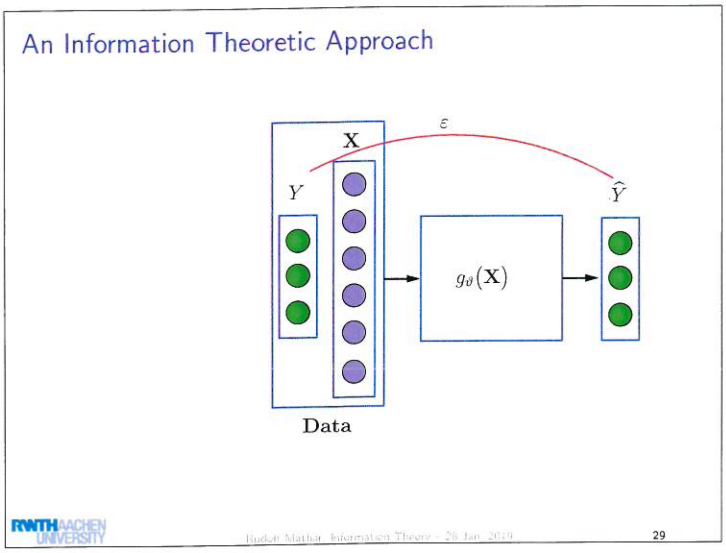
$$\mathcal{S} = \big\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\big\}$$

Training error

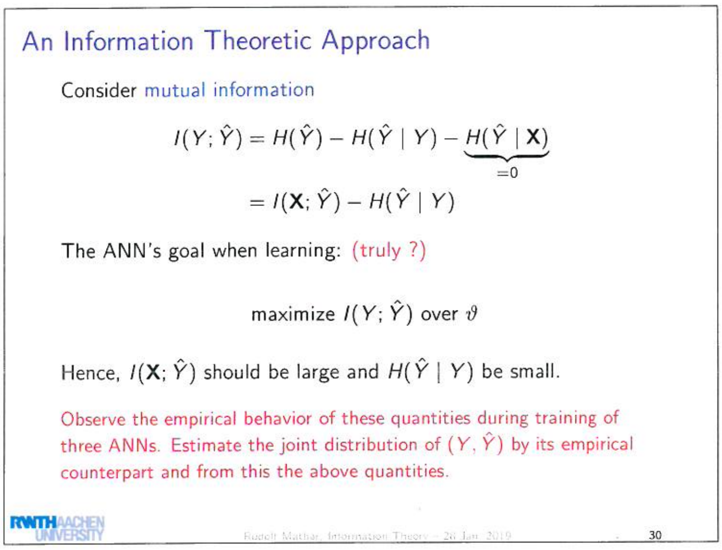
$$\hat{R}(g_\vartheta) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\big(g_\vartheta(\mathbf{x}_i) \neq y\big) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(\hat{y} \neq y)$$

Overview

- Basic concepts from information theory
- Biological neural networks (BioNN)
- Artificial neural networks (ANN)
- An information theoretic approach
- Noisy training data
- Outlook



An Information Theoretic Approach



An Information Theoretic Approach

Consider mutual information

$$I(Y;\hat{Y}) = H(\hat{Y}) - H(\hat{Y} \mid Y) - \underbrace{H(\hat{Y} \mid \mathbf{X})}_{=0}$$

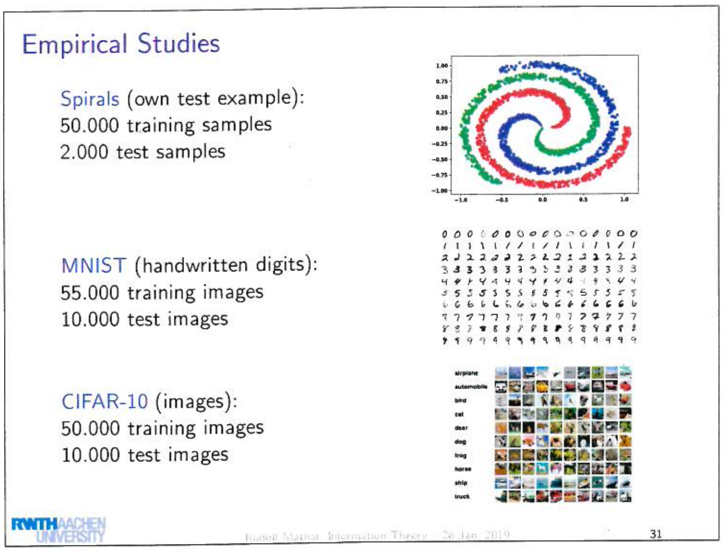
$$= I(\mathbf{X};\hat{Y}) - H(\hat{Y} \mid Y)$$

The ANN's goal when learning: (truly ?)

maximize $I(Y;\hat{Y})$ over ϑ

Hence, $I(\mathbf{X};\hat{Y})$ should be large and $H(\hat{Y} \mid Y)$ be small.

Observe the empirical behavior of these quantities during training of three ANNs. Estimate the joint distribution of $(Y,\hat{Y})$ by its empirical counterpart and from this the above quantities.

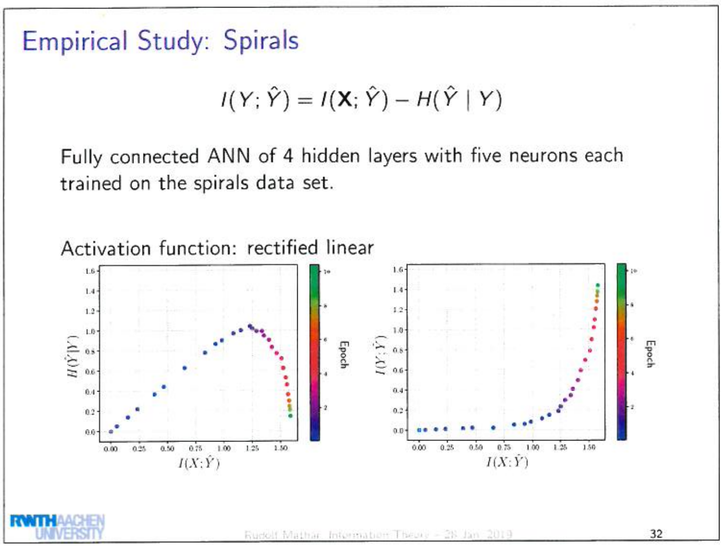


Empirical Studies

Spirals (own test example):
50.000 training samples
2.000 test samples

MNIST (handwritten digits):
55.000 training images
10.000 test images

CIFAR-10 (images):
50.000 training images
10.000 test images

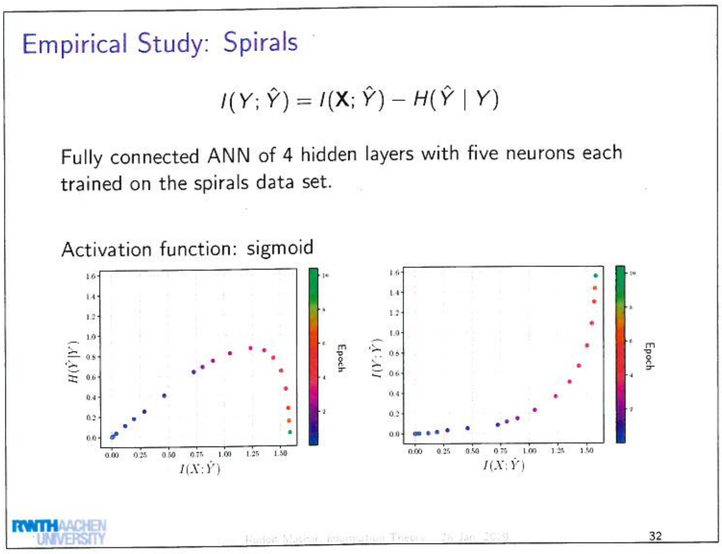


Empirical Study: Spirals

$$I(Y;\hat{Y}) = I(\mathbf{X};\hat{Y}) - H(\hat{Y} \mid Y)$$

Fully connected ANN of 4 hidden layers with five neurons each trained on the spirals data set.

Activation function: rectified linear

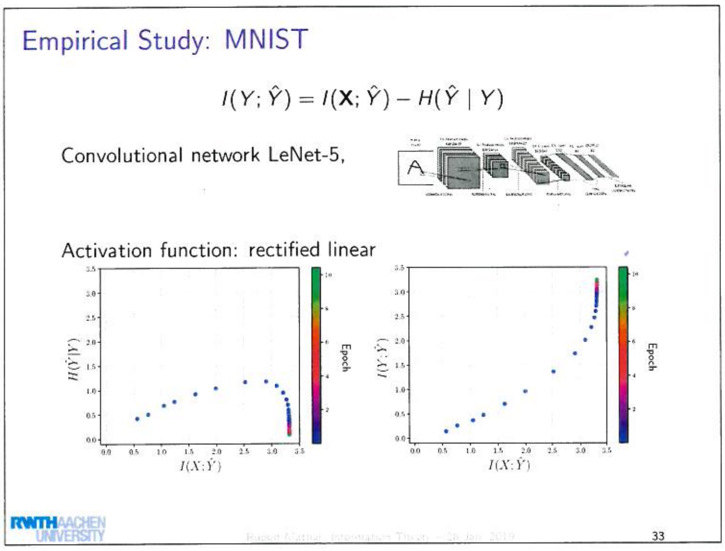


Empirical Study: Spirals

$$I(Y;\hat{Y}) = I(\mathbf{X};\hat{Y}) - H(\hat{Y} \mid Y)$$

Fully connected ANN of 4 hidden layers with five neurons each trained on the spirals data set.

Activation function: sigmoid

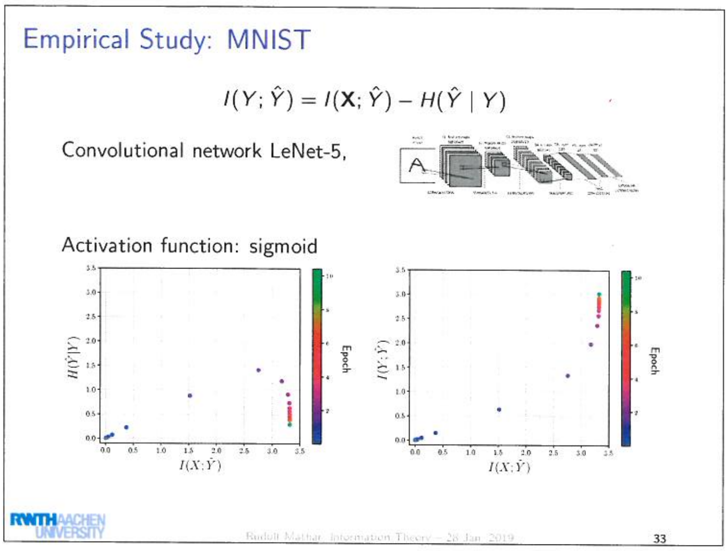


Empirical Study: MNIST

$$I(Y;\hat{Y}) = I(\mathbf{X};\hat{Y}) - H(\hat{Y} \mid Y)$$

Convolutional network LeNet-5,

Activation function: rectified linear

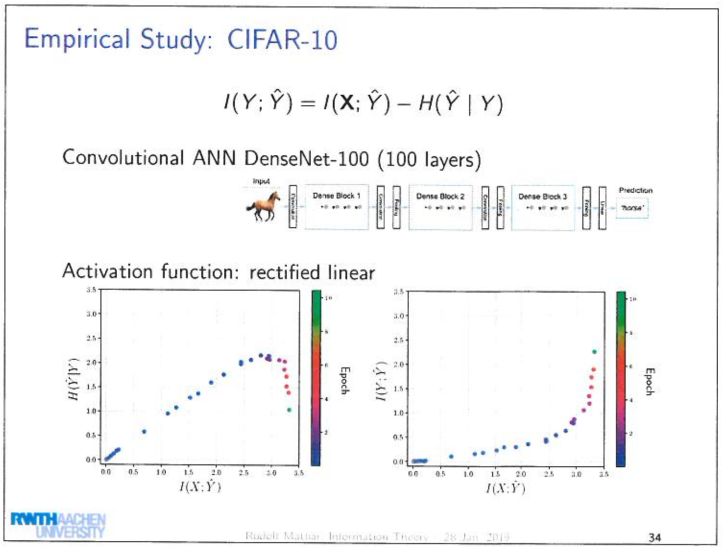


Empirical Study: MNIST

$$I(Y;\hat{Y}) = I(\mathbf{X};\hat{Y}) - H(\hat{Y} \mid Y)$$

Convolutional network LeNet-5,

Activation function: sigmoid



Empirical Study: CIFAR-10

$$I(Y;\hat{Y}) = I(\mathbf{X};\hat{Y}) - H(\hat{Y} \mid Y)$$

Convolutional ANN DenseNet-100 (100 layers)

Activation function: rectified linear

Expected Error and Conditional Entropy

Recall the expected error or test error

$$\varepsilon = R(g_\vartheta) = P(\hat{Y} \neq Y)$$

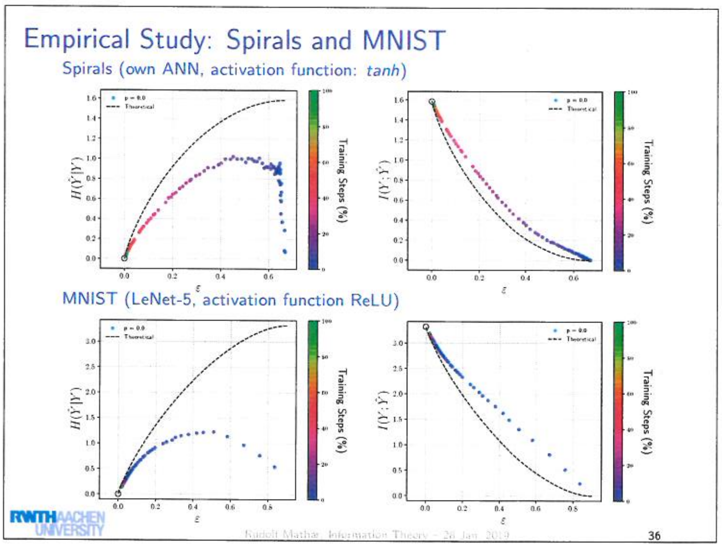
Theorem

$$\max\left\{H(\hat{Y} \mid Y), H(Y \mid \hat{Y})\right\} \leq \Psi(R(g_\vartheta))$$

where

$$\Psi(x) = x\log(K-1) - x\log(x) - (1-x)\log(1-x).$$

Proof. Fano's inequality.



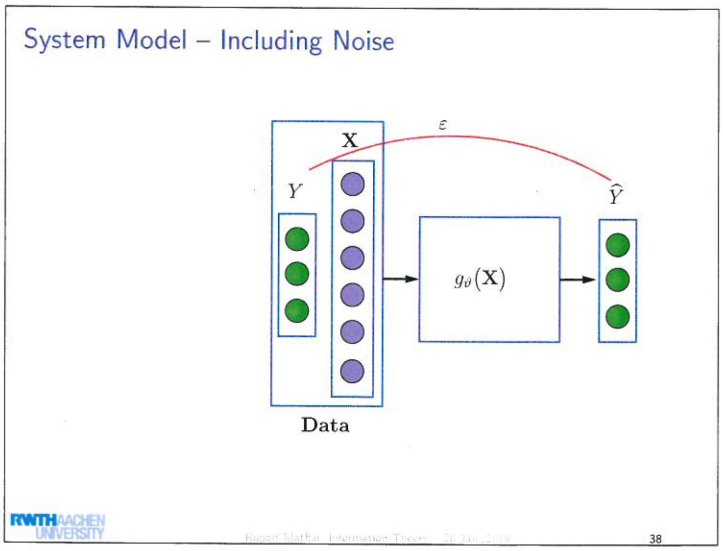
Empirical Study: Spirals and MNIST

Spirals (own ANN, activation function: *tanh*)

MNIST (LeNet-5, activation function ReLU)

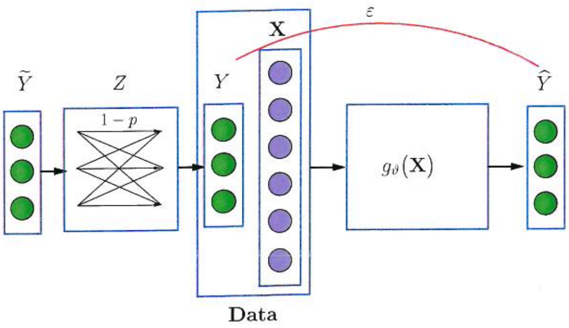
Overview

- Basic concepts from information theory
- Biological neural networks (BioNN)
- Artificial neural networks (ANN)
- An information theoretic approach
- Noisy training data
- Outlook



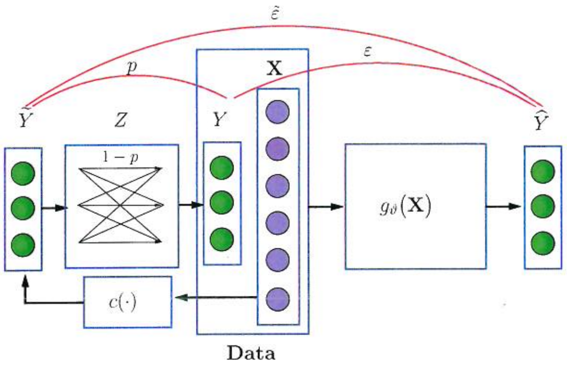
System Model – Including Noise

System Model – Including Noise



The expert determining class labels may be error prone.

System Model – Including Noise



The expert determining class labels may be error prone.

Noisy Training Data

Assume $\tilde{Y} = c(\mathbf{X}) \in \{0, 1, \dots K-1\}$ is the true class label of $\mathbf{X}$. The observed class label is noisy as

$$Y = (\tilde{Y} + B) \bmod K = \tilde{Y} \oplus B,$$

with B an independent random variable attaining values in $\{0, 1, \dots K-1\}$.

Theorem

Assume that a neural network is trained with noisy labels Y. Then

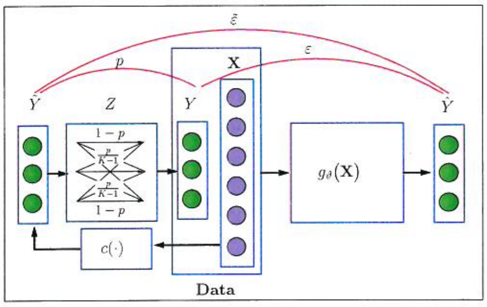
$$\varepsilon = R(g_\vartheta) \geq \Phi(H(B))$$

where $\Phi(\cdot)$ is the inverse of Ψ on $[0, 1 - \frac{1}{K}]$.

Noisy Training Data

Special case:

$$P(B = i) = \begin{cases} 1-p, & \text{if } i = 0 \\ \frac{p}{K-1}, & \text{if } i = 1, \dots, K-1 \end{cases}$$



Noisy Training Data

In this case,

$$H(B) = h_{bin}(p) + p\log(K-1) = \Psi(p),$$

hence,

$$R(g_\vartheta) \geq \Phi(\Psi(p)) = p$$

Noisy Training Data

In this case,

$$H(B) = h_{bin}(p) + p\log(K-1) = \Psi(p),$$

hence,

$$R(g_\vartheta) \geq \Phi(\Psi(p)) = p$$

and as a surprise ...

Theorem

Let $Y = \tilde{Y} \oplus B$ and $p < 1 - \frac{1}{K}$. If $R(g_\vartheta) = P(\hat{Y} \neq Y) = p$ (the lower bound for the test error is achieved) then

$$P(\hat{Y} = \tilde{Y}) = 1.$$

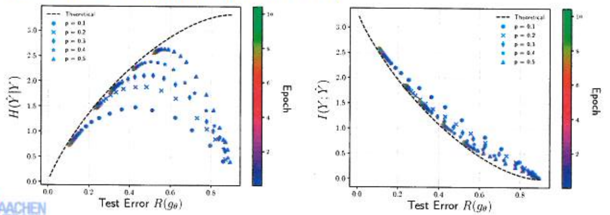
Noisy Training Data

Theorem

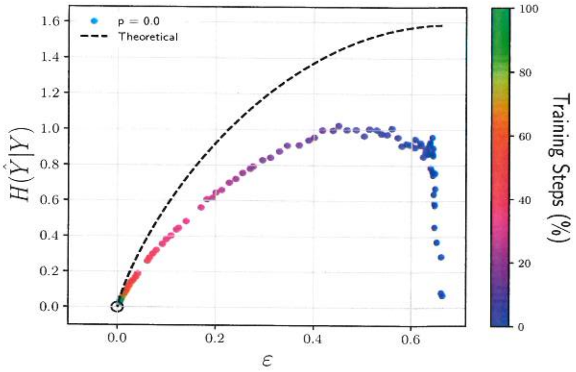
Let $\tilde{Y}$ be uniformly distributed. If $R(g_\vartheta) = P(\hat{Y} \neq Y) = p$, then

$$I(\mathbf{X};\hat{Y}) = H(\hat{Y}) = \log K$$
$$H(\hat{Y} \mid Y) = \Psi(p)$$
$$I(Y;\hat{Y}) = \log K - \Psi(p)$$

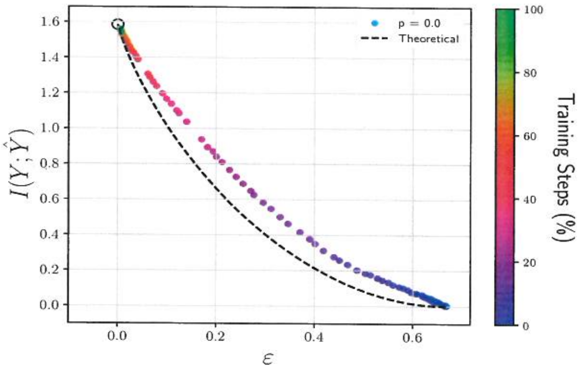
MNIST (LeNet-5, activation function ReLU)



Noisy Training Data - Animation



Noisy Training Data - Animation



Overview

- Basic concepts from information theory
- Biological neural networks (BioNN)
- Artificial neural networks (ANN)
- An information theoretic approach
- Noisy training data
- Outlook

Outlook – Markovian Structures

Parameters ϑ_n during training epochs with pure gradient descent form a Markov chain (MC).

$$\vartheta_n = f(\vartheta_{n-1}, \mathbf{T}_n) \text{ with i.i.d. training data } \mathbf{T}_n.$$

However,

$$\hat{Y}_n^{(c)} = g(\vartheta_n, \mathbf{X}),\ n \in \mathbb{N}$$

is not a MC because of strong coupling. Consider instead a decoupled version

$$\hat{Y}_n = g(\vartheta_n, \mathbf{X}_n) \text{ with i.i.d. input data } \mathbf{X}_n.$$

$\hat{Y}_n$ is a hidden Markov chain (HMC) with

$$H(\hat{Y}_n) = H(\hat{Y}_n^{(c)}).$$

Outlook – Markovian Structures

Theorem

Let the MC $\vartheta_n \sim \mathbf{q}_n$ have stationary distribution $\mathbf{q}$. Then

$$\log K \geq H(\hat{Y}_n) \geq \log K - D(\mathbf{q}_n \| \mathbf{q}).$$

Outlook – Markovian Structures

Theorem

Let the MC $\vartheta_n \sim \mathbf{q}_n$ have stationary distribution $\mathbf{q}$. Then

$$\log K \geq H(\hat{Y}_n) \geq \log K - D(\mathbf{q}_n \| \mathbf{q}).$$

Digging deeper into the Markovian structure of ANNs ...

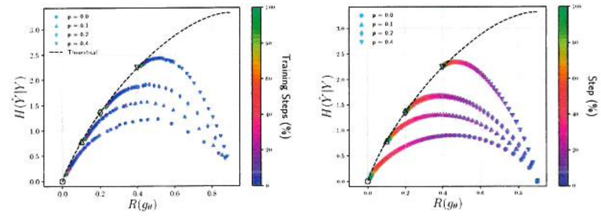
Consider the joint distribution of $(\tilde{Y}_n, \hat{Y}_n)$ as marginal distribution of a MC $\mathbf{Z}_n = (\tilde{Y}_n, \hat{Y}_n)$.

$$P^{\mathbf{Z}_0} = \boxed{\begin{array}{cccc} \frac{1}{K} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ \frac{1}{K} & 0 & \cdots & 0 \end{array}} \Rightarrow \cdots \Rightarrow \boxed{\begin{array}{ccc} \frac{1}{K} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{K} \end{array}} = P^{\mathbf{Z}_\infty}$$

Construct a 2-dim MC with the LHS as initial and the RHS as limit distribution.

Outlook – Markovian Structures

MNIST (LeNet-5, activation function ReLU)



Experiments

Markov Process

Thanks for your attention!

balda/behboodi/mathar@ti.rwth-aachen.de

Quotations

Graham Taylor about developing ANN mimicking visual abilities of drosophila:

"The approach of pairing deep learning models with nervous systems is incredibly rich. It can tell us about the models, about how neurons communicate with each other, and it can tell us about the whole animal. That's sort of mind blowing. And it's unexplored territory."

https://www.cifar.ca/cifarnews/2018/10/25/building-a-fly-brain-in-a-computer