Deep Learning

Introductory Lecture

Tuesday 15th October 2019

Dr. Nicholas Cummins

Introductory Lecture

Welcome!

What is it?

Who are we?

Deep Learning

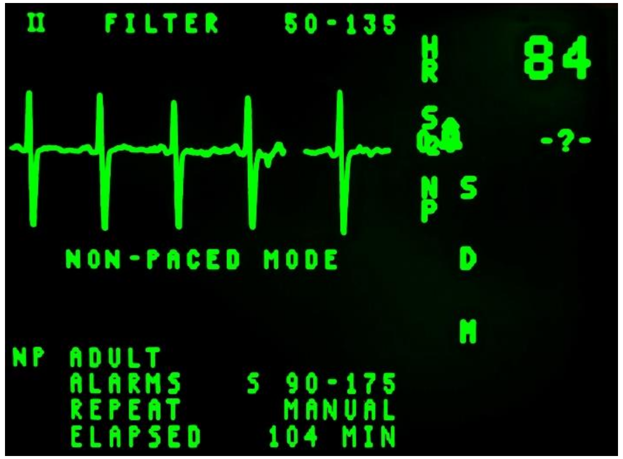
What will we do?

Why?

What do we expect?

Welcome

Images Sourced: https://pixabay.com



Universität Augsburg University

Welcome!

What is it?

Who are we?

Deep Learning

What will we do?

Why?

What do we expect?

Contacts

- **Prof. Björn Schuller**
  - bjoern.schuller@informatik.uni-augsburg.de
- **Dr. Nicholas Cummins**
  - nicholas.cummins@informatik.uni-augsburg.de
- **Manuel Milling**
  - manuel.milling@informatik.uni-augsburg.de

Universität Augsburg University

Welcome!

What is it?

Who are we?

Deep Learning

What will we do?

Why?

What do we expect?

The AI/ML/DL Relationship

Artificial Intelligence:

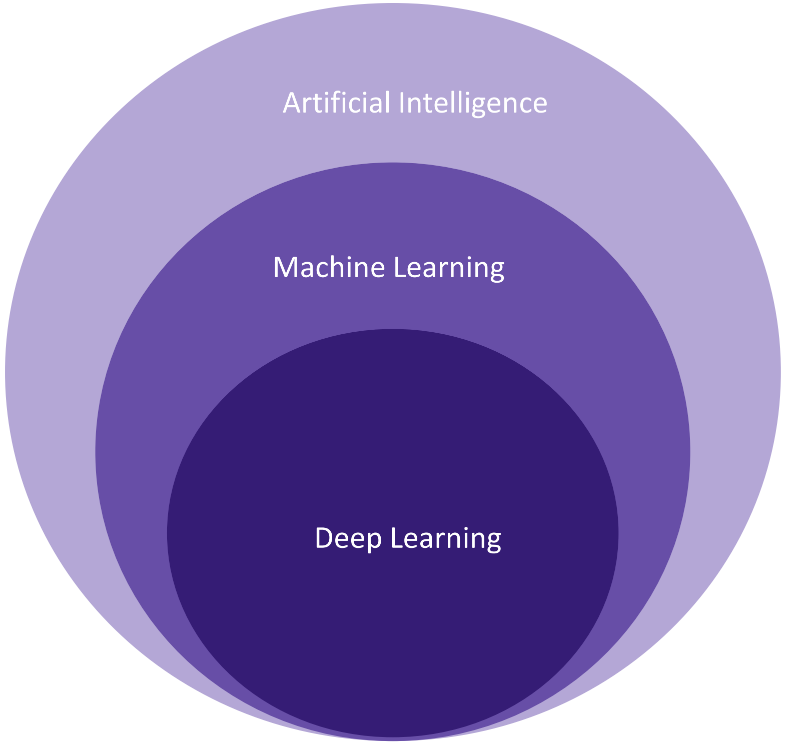
- A broad concept where machines think and act more like humans

Machine Learning:

- An application of AI where machines use date to automatically improve at performing tasks

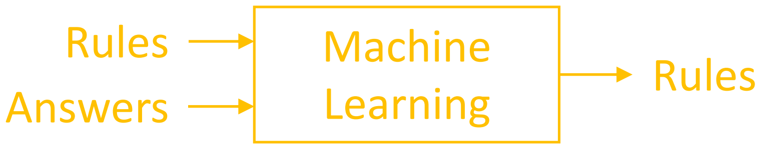
Deep Learning:

- A machine learning technique that processes data through a multi-layered neural network much like the human brain



What is Machine Learning?

- **Machine Learning**
  - **Discovering rules to execute a data-processing task**
  - **Classic programming:**
    - Input rules (a program) and data
    - Process the data according to these rules
    - Output answers
  - **Machine Learning:**
    - Input data as well as the answers expected from the data
    - Learning the rules need to map between data and answers
    - Output the rules so they can then be applied to new data to produce original answers

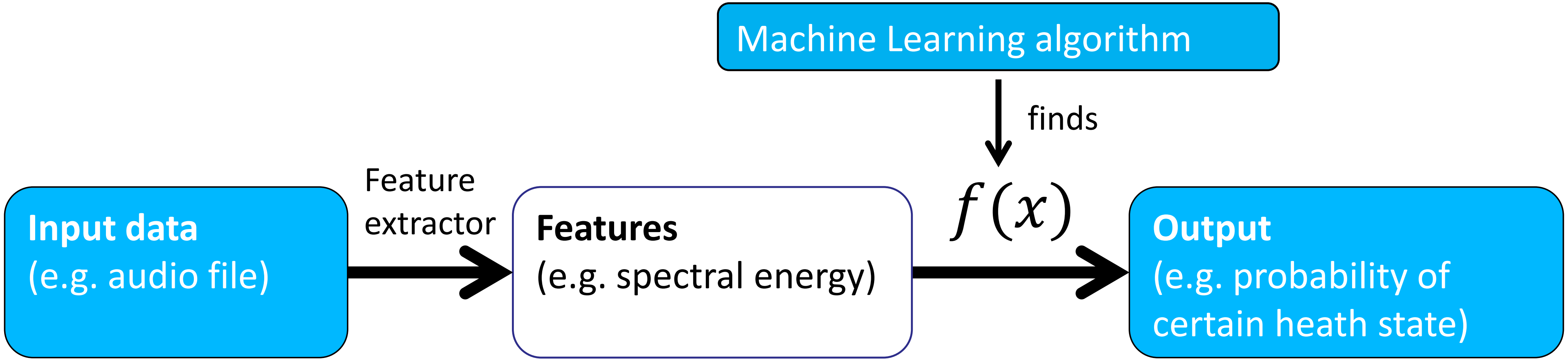


- **Machine Learning**
  - **Discovering rules to execute a data-processing task**
  - A machine-learning system is trained rather than explicitly programmed.
    - It's presented with many examples relevant to a task
    - It identifies statistical structure in these examples
    - These structure eventually allows the system to determine rules for automating the task
  - Unlike optimisation and conventional statistical analysis we want to learn rules that are generalisable to new data instances

- **What is needed to do machine learning?**
  1. **Input data points**
     - E.g. for speech recognition, the data should be sound files of people speaking.
  2. **Examples of the expected output**
     - E.g. for speech-recognition, human-generated transcripts of sound files
  3. **A way to measure whether the algorithm is doing a good job**
     - This is necessary in order to determine the distance between the algorithm's current output and its expected output.
     - The measurement is used as a feedback signal to adjust the way the algorithm works.
     - **This adjustment step is the system learning**

Typical processing chain

- Feature extractor
  - Extracts useful pieces of information from raw data
- Machine learning algorithm
  - Learn rules to predict labels to features



Universität Augsburg University

What are features?

- The representation of the data presented to the machine learning algorithm
- Each feature can be thought of as a single piece of information the algorithm can use when making a decision
- Typically hundreds or thousands of such pieces of information are concatenated together to form a feature vector
- The role of the machine learning algorithm is to identify patterns from a collection of feature vectors

Machine Learning Algorithms

- Creation of (robust) models to predict/classify a particular output (y) from a selected independent variables (X – features) from a dataset
  - Primarily concerned with the identification of patterns within (large amounts of) data
  - Machine learning algorithms are used to perform the process of pattern identification via an iterative process
  - Learning phase: the algorithm optimises its parameters with the goal of improving (recognition) performance on a particular task
  - Deep learning is a particular form of machine learning algorithms

Universität Augsburg University

Goal

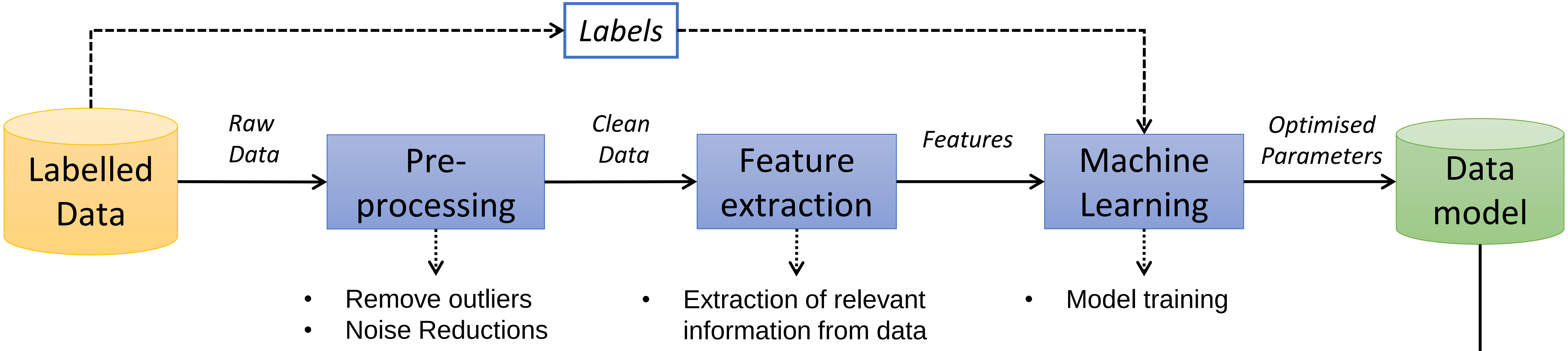
- Learn a *robust* predictive function $f(\cdot)$
- A mapping from the feature space $\mathcal{X}$ to the label space $\mathcal{Y}$

$$\mathcal{X} \xrightarrow{f(\cdot)} \mathcal{Y}$$

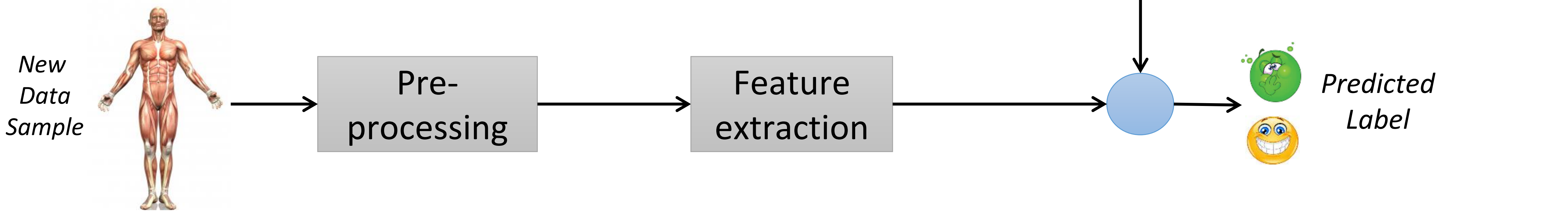
- Given a test sample (unknown label), the learnt function maps the test feature vector $\boldsymbol{x}_*$ into a specific label $\boldsymbol{y}_*$

$$\boldsymbol{y}_* = f(\boldsymbol{x}_*)$$

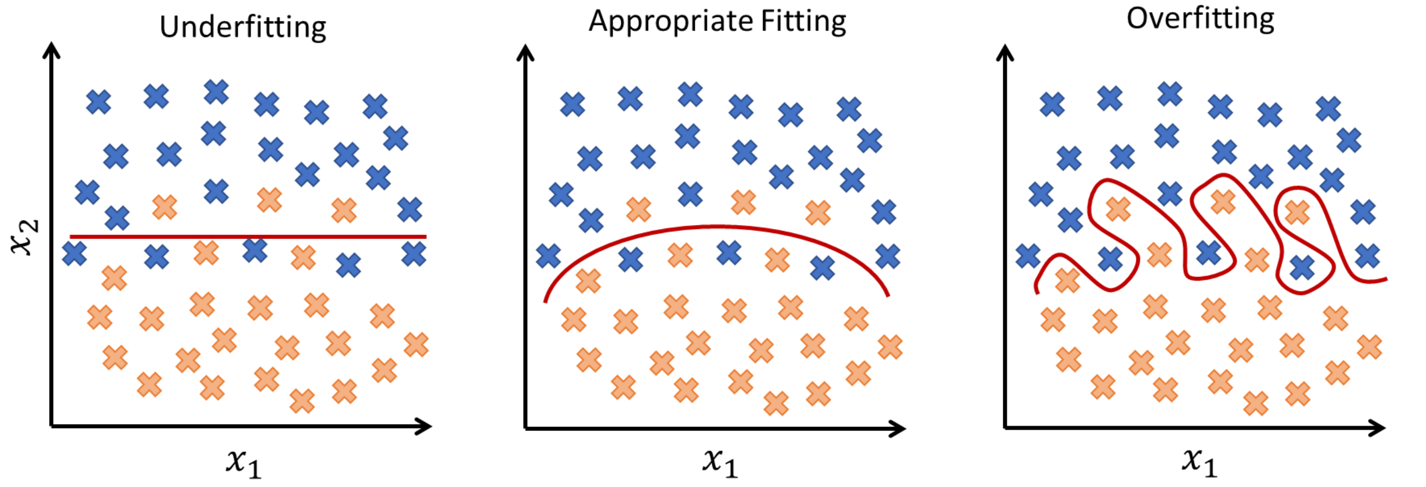
- **Training phase** – *minimise training errors*



- **Test phase** – *minimise generalisation errors*

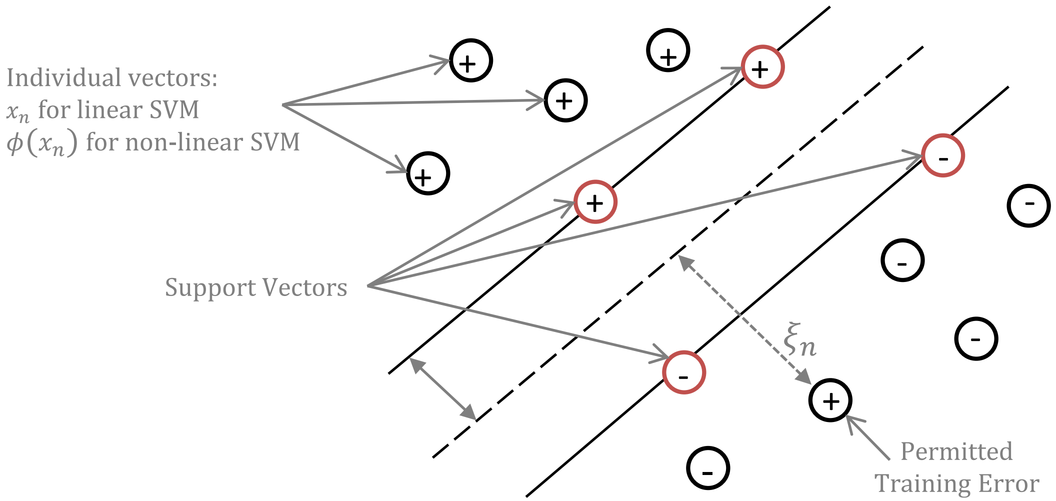


- **Generalisation Errors**
  - **Underfitting** – the model is too simple
    - The model has high bias and lacks sensitivity to the variation in data
  - **Overfitting** – the model is too complex
    - Model attempts to account for all the variation in the training data

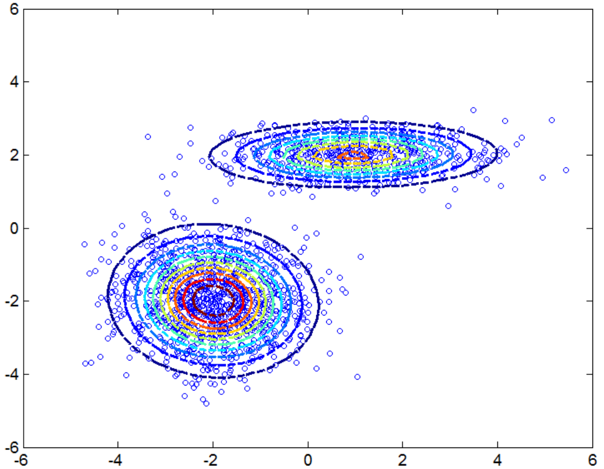


What is Machine Learning?

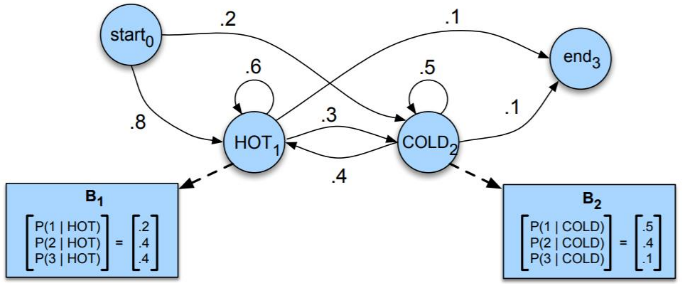
Support Vector Machines



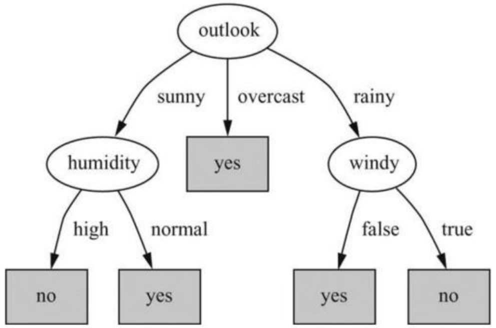
Gaussian Mixture Models



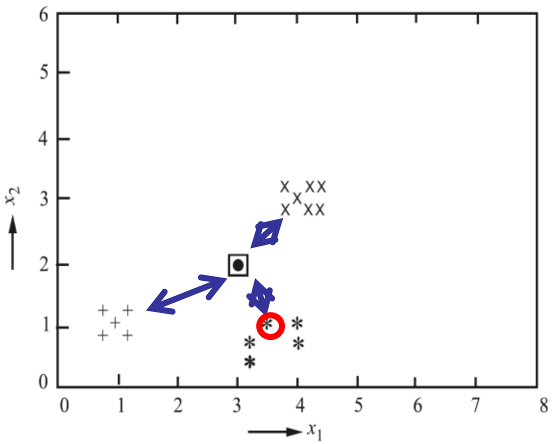
Hidden Markov Models



Decision Trees



K-means

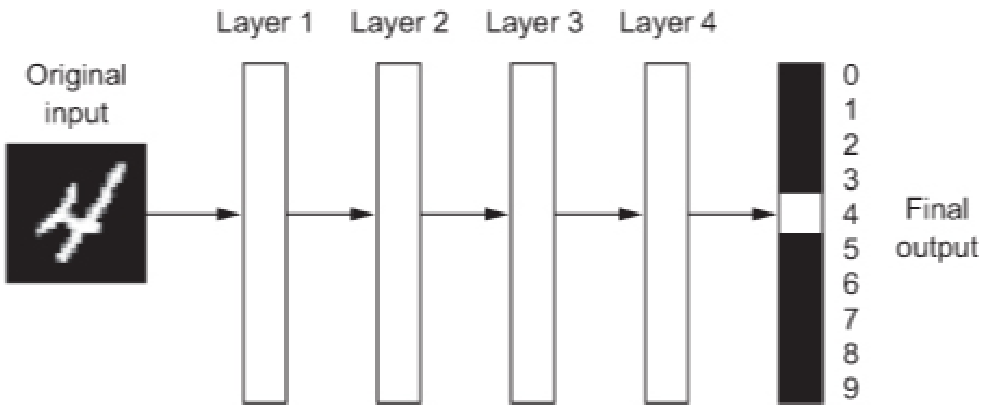


- **Data-Transformation**
  - A machine-learning model transforms its input data into meaningful outputs

$$x \xrightarrow{f(\cdot)} y$$

  - This process that is “learned” from exposure to known examples of inputs and outputs
  - Learning to meaningfully transfer data is the central problem in machine learning and deep learning
    - Learn useful representations of the input data at hand
    - These representations should get us closer to the expected output

- **Deep learning is a specific subfield of machine learning**
  - Algorithms that put specific emphasis on learning successive layers of meaning full representations
  - The term deep represents this idea of successive layers of representations

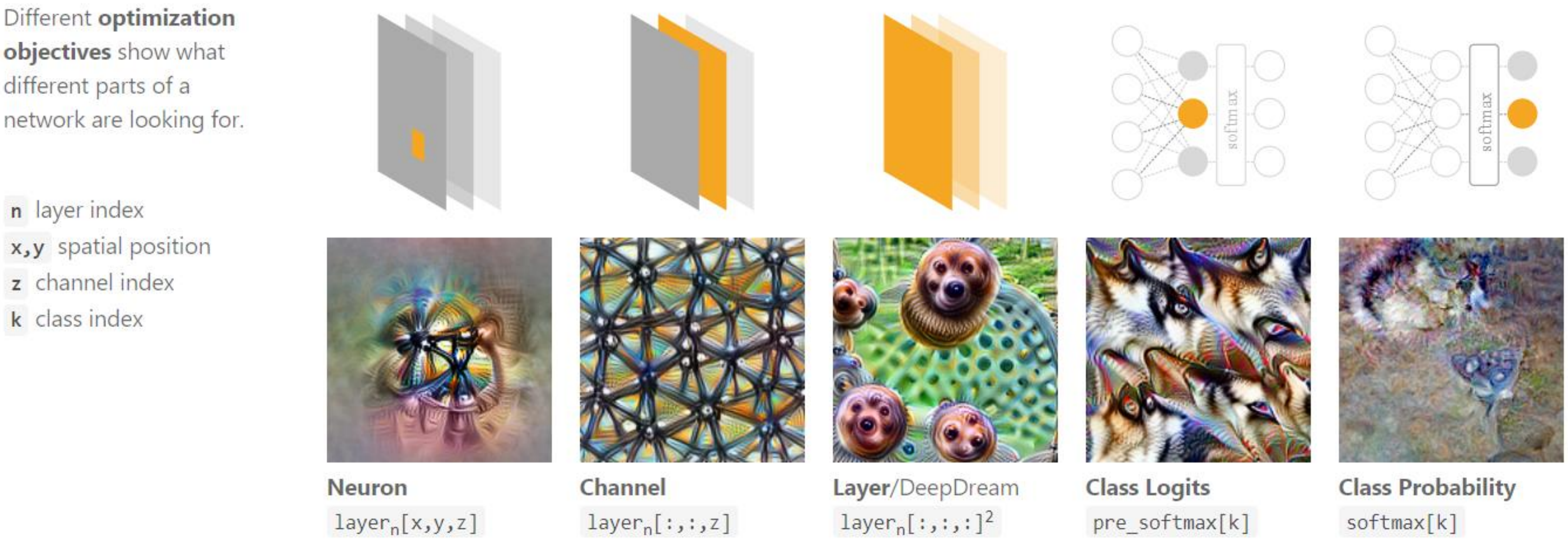


How many layers contribute to a model of the data is called the *depth* of the model

What is Deep Learning?

Image Source: https://distill.pub/2017/feature-visualization/

- **Deep learning is a set of multistage techniques for learning successive data representations**
  - A DNN transforms input data into a set of representations that are increasingly informative about the final result



History of Deep Learning

- **1943 McCulloch+Pitts**
  - 1st formal computer model based on the neural networks of the human brain
- **1949 Hebb**
  - Synaptic connection between 2 neurons is enhanced by frequent activation (Hebb rule)
  - Led to methodology for determining how to alter the weights between model neurons
- **1958 Rosenblatt**
  - 1st Neurocomputer-Perceptron (mechanical)
- **1960 Widrow+Hoff**
  - *Least Mean Square* training algorithm
  - Introduced the *Adaptive Linear Neuron* (Adaline) network
- **1965 Vakhnenko+Lapa**
  - Learning algorithm for supervised deep feedforward multilayer perceptrons
- **1969 Minsky+Papert**
  - Demonstrate limits of single layer multilayer perceptrons
  - The XOR problem
- **1962 Dreyfus**
  - Backpropagation based on the chain rule only
- **1970 Linnainmaa**
  - Backpropagation for discrete sparse networks

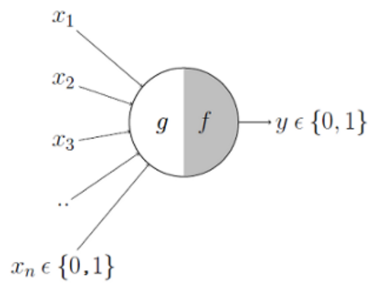


Image Source: https://towardsdatascience.com/

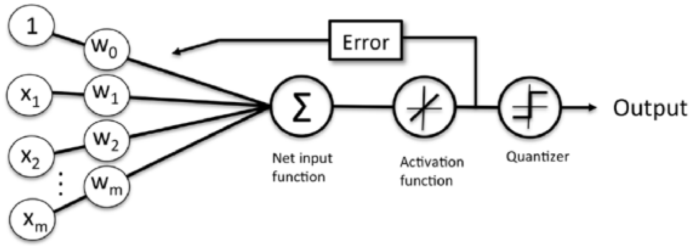


Image Source: https://www.bogotobogo.com

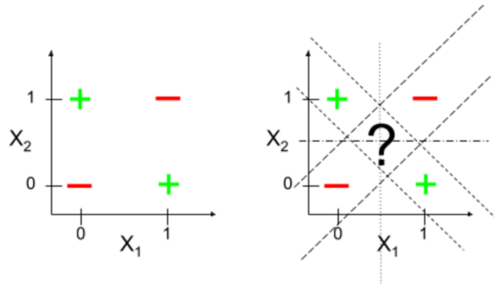
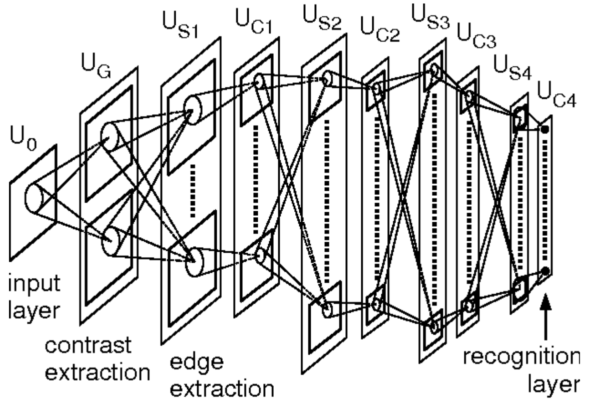


Image Source: www.tech-quantum.com

History of Deep Learning

- **1974 to 1980 – ‘The First AI Winter’**
  - Initial promises of AI not meet, long training time and large computational resources
  - The research arm of the U.S. Defense Department cut its funding of AI researchers
  - Winter ended with the advent of expert, i.e., domain specific, systems
- **1980 Fukushima**
  - Proposed the *Neoconitron*, a hierarchical, multilayered artificial neural network
  - Precursor to modern Convolutional nets (CNNs)
- **1986 Rumelhart+Hinton+Williams:**
  - Published “Learning Representations by Back-propagating Errors”
  - Backpropagation in n-layer models
- **1987 to 1993 – ‘The Second AI Winter’**
  - Again, promises of AI not meet, training times and computational resources still an issue
- **1987 Ballard**
  - Proposed autoencoders as a method for unsupervised pre-training
- **1989 LeCun**
  - First demonstration of backpropagation applied to CNNs
- **1991 Hochreiter+Schmidhuber**
  - Demonstrate that deep feedforward or recurrent networks are hard to train by backpropagation
  - The vanishing and exploding gradient issues



Fukushima, K., 2003. Neocognitron for handwritten digit recognition. Neurocomputing, 51, pp.161-180.

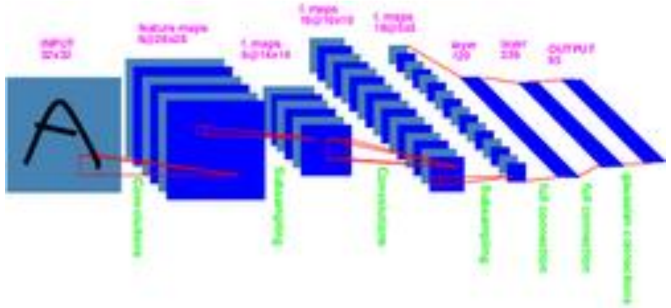


Image Source: http://yann.lecun.com

History of Deep Learning

- **1997 Hochreiter+Schmidhuber**
  - Proposed the Long Short-Term Memory RNN
  - Arguably the first purely supervised deep learner

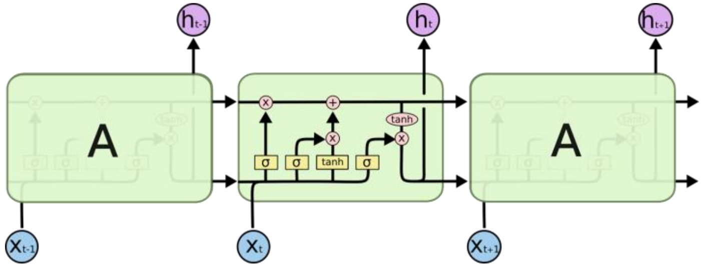


Image source: https://colah.github.io/

- **2006 Hinton+Salakhutdinov**
  - Unsupervised pre-training of feedforward NN - Deep Belief Networks

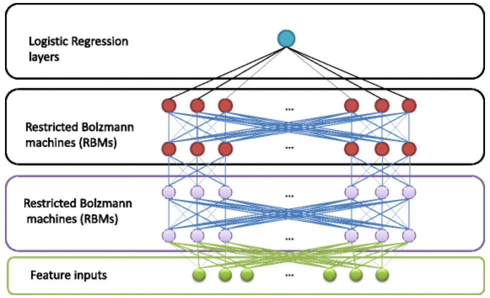


Image Source: https://medium.com

- **2007 – start of the GPU era**
  - Rivalry between Nvidia and ATI lead to considerable advances in GPU capabilities
- **2009 Increasing dataset sizes**
  - Release of ImageNet Data set, a free database of more than 14 million labelled image
- **Late 2000's early 2010's**
  - Numerous demonstrations of direct training of large DNN's on GPU clusters
- **2010 Nair+Hinton**
  - Propose the Rectified Linear Unit (ReLu) activation function
- **2012 Krizhevsky+Hinton**
  - AlexNet, a GPU-implemented CNN wins ImageNet Challenge
- **2014 Srivastava *et al***
  - Propose dropout as a simple method for preventing overfitting
- **2014 Goodfellow *et al***
  - Propose Generative Adversarial Networks for data augmentation

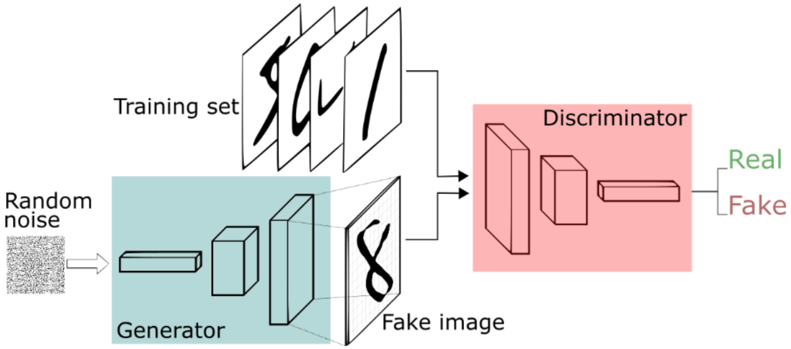
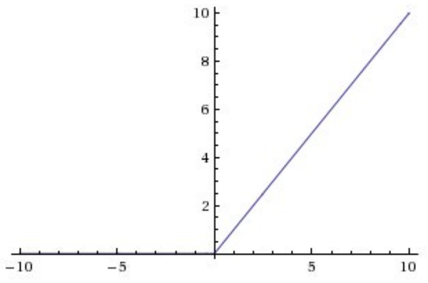


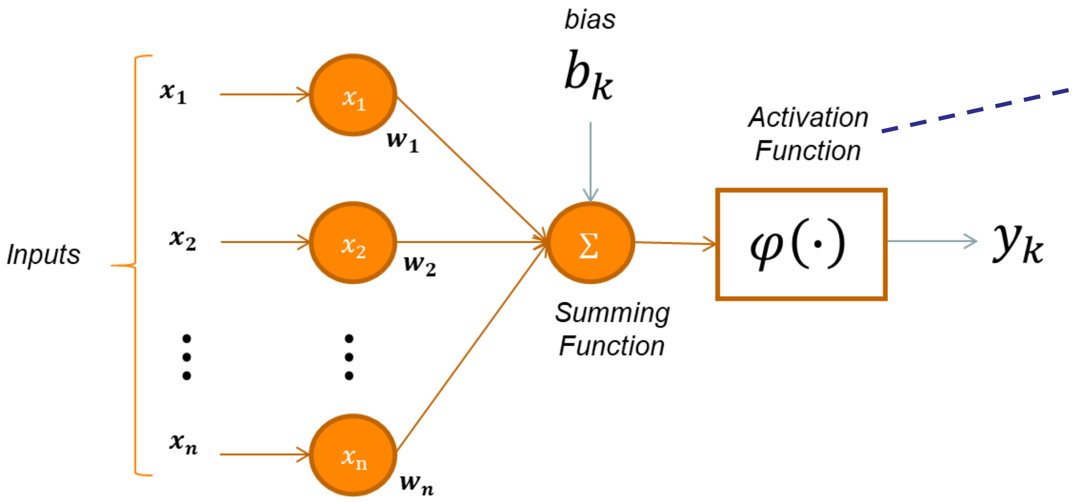
Image Source: https://skymind.ai/

Artificial Neurons

- Building block of neural networks
  - Combines different inputs to make a single output
- Activations Function
  - Inclusion of nonlinearities
  - Enable learning of complex patterns



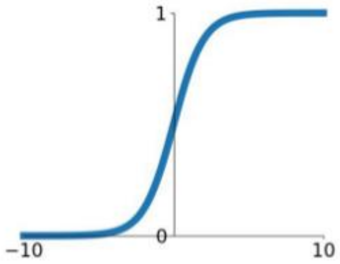
Rectified Linear unit
$f(x) = \max(0, x)$



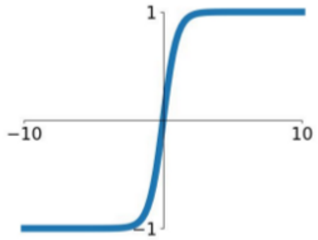
$$y_k = \varphi\left(\begin{pmatrix} w_1 & w_2 & \dots & b \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \dots \\ 1 \end{pmatrix}\right) = \varphi(w_1 x_1 + w_2 x_2 + \cdots + b\,)$$

Activation Functions

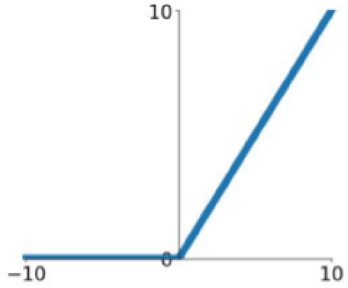
- Uses the weighted input value to determine the level of output activation
  - Introduces nonlinearities into network
- Typical activation functions include
  - Identity → $f(x) = x$
  - Logistic → $f(x) = 1/(1 + \exp(-x))$
  - Tanh → $f(x) = \tanh x$
  - Rectified Linear unit → $f(x) = \max(0, x)$
  - Sigmoid → $f(x) = 1/(1 + \exp^{x})$



Sigmoid

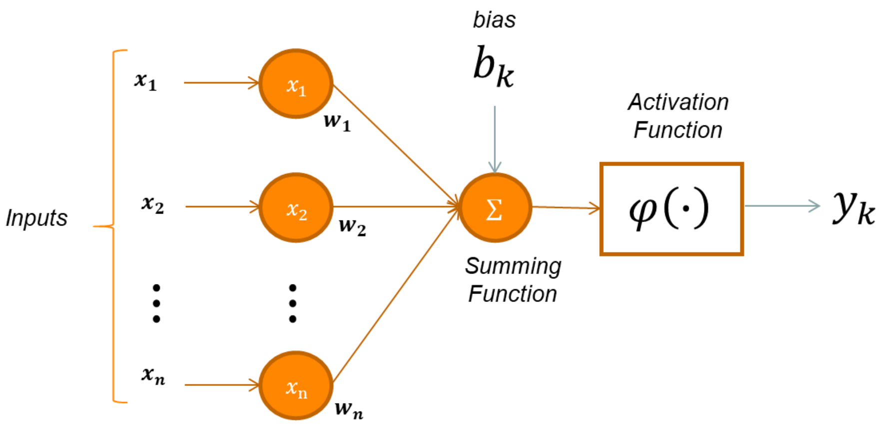


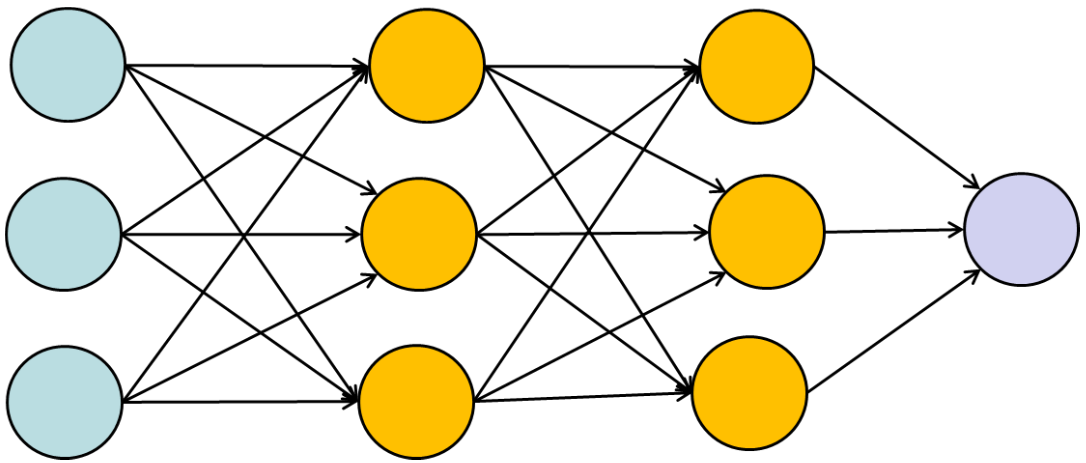
Tanh



ReLU

- **Neural networks**
  - In deep learning, the layered representations are (almost always) learned via models called neural networks structured in literal layers stacked on top of each other





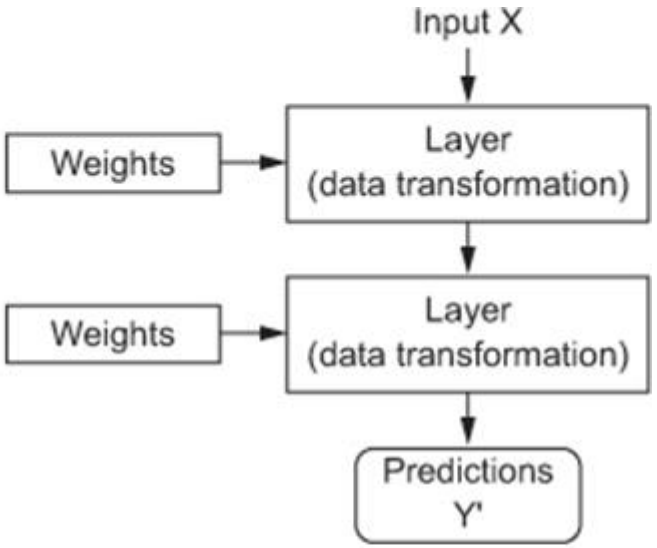
$$\varphi\left(\begin{pmatrix} w_1 & w_2 & \dots & b \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ \dots \\ 1 \end{pmatrix}\right) = \varphi(w_1x_1 + w_2x_2 + \cdots + b) = y_k$$

Universität Augsburg University

- **Weights Learnt via Gradient Descent**
  - Update weights to minimise loss function
  - This is achieved by taking the gradient of loss function with respect to the weights

$$W += W + \alpha \frac{\partial j}{\partial w}$$

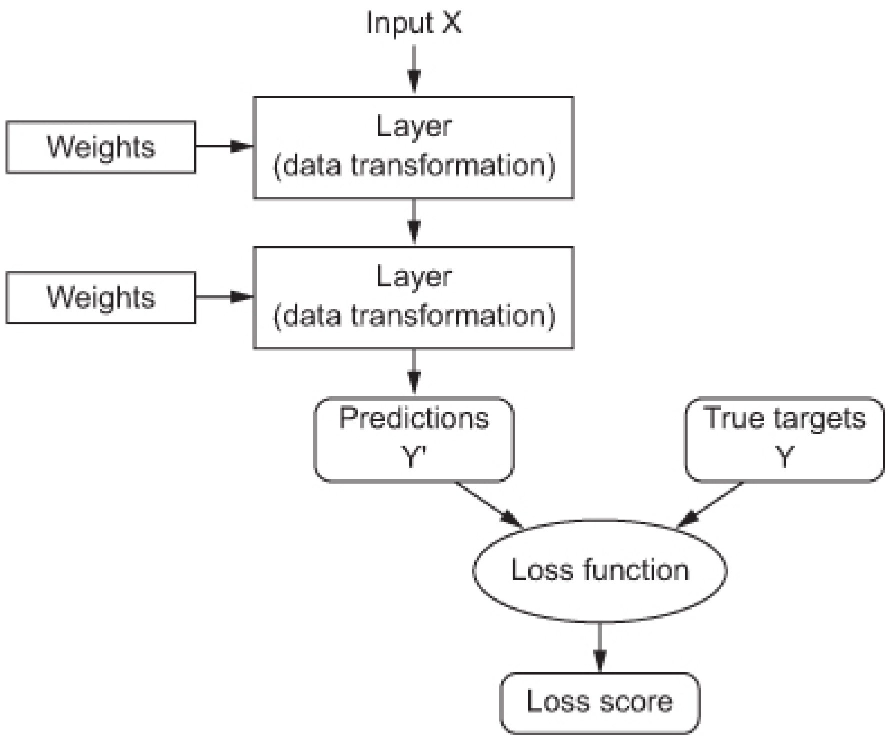
  - Not a trivial process as neural networks are structured as a series of layers
  - A single network can contain many millions of weights, and modifying the value of one weight will affect the behaviour of all the others



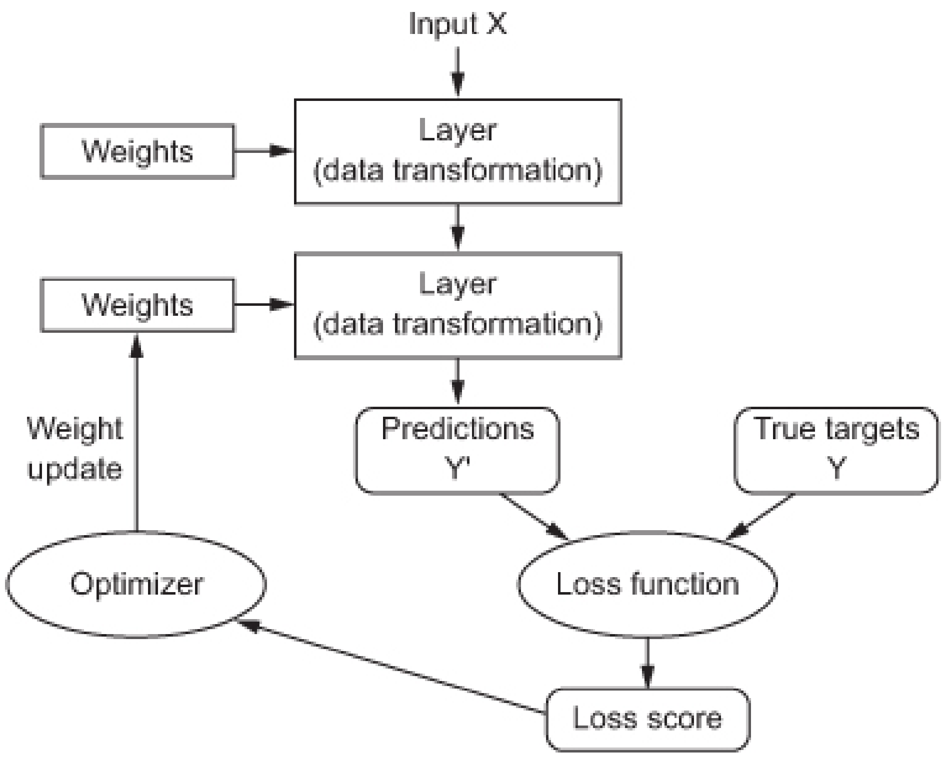
- **Learning in Deep Neural Networks**
  - To control weight updates in neural networks we use a loss function to how far an output prediction is from what we expected
    - The loss function computes a single scalar value relating to network performance
    - Measures the difference between what we have predicted, $\tilde{y}$, with the what it should predicted y.

$$\mathcal{L}(\tilde{y}, y)$$

    - We can then use this information to update the network



- **Learning in Deep Neural Networks**
  - The loss function provides a feedback signal to adjust the weights by a small amount, in a particular direction that will lower the score
  - This adjustment is performed by an *optimizer*, which implements the *Backpropagation* algorithm
    - Error attribution: figuring out how much each weight contributed to the final error by propagating the error back through the network



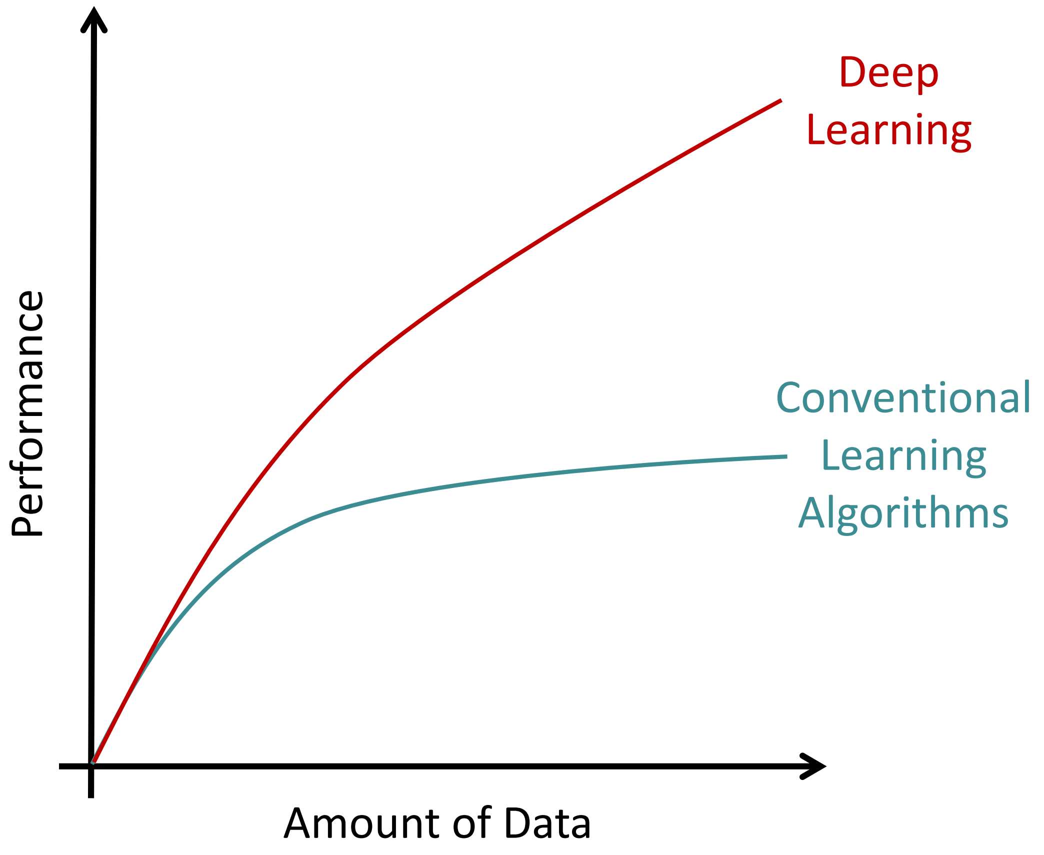
Universität Augsburg University

Performs well on large datasets

- Conventional algorithms do not scale well to huge amounts of data
  - Especially true in relation to complex problems in image classification, natural language processing, and speech processing

Universality

- A network with a single sufficiently large hidden layer (in theory) is adequate for the approximation of most functions

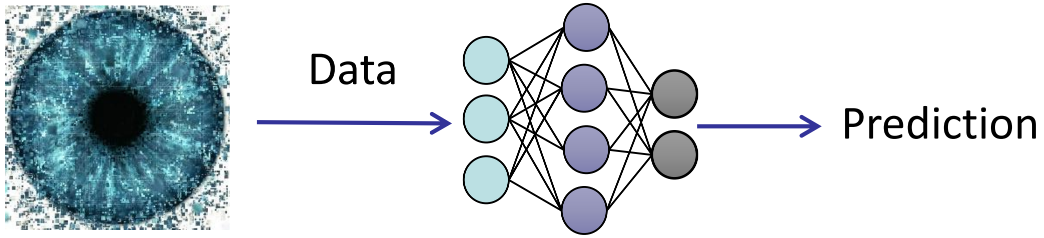


Why use Deep Learning?

End to End Processing

- Conventional Machine learning relies of features to reduce data complexity & make patterns more visible

- Deep Learning algorithms learn high-level features from data in an incremental manner.
  - Eliminates the need of domain expertise and feature extraction.

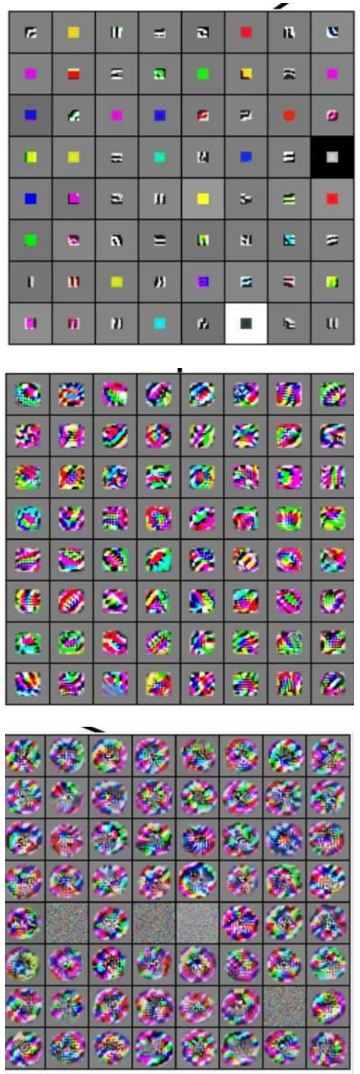


Images Sourced: https://pixabay.com

Universität Augsburg University

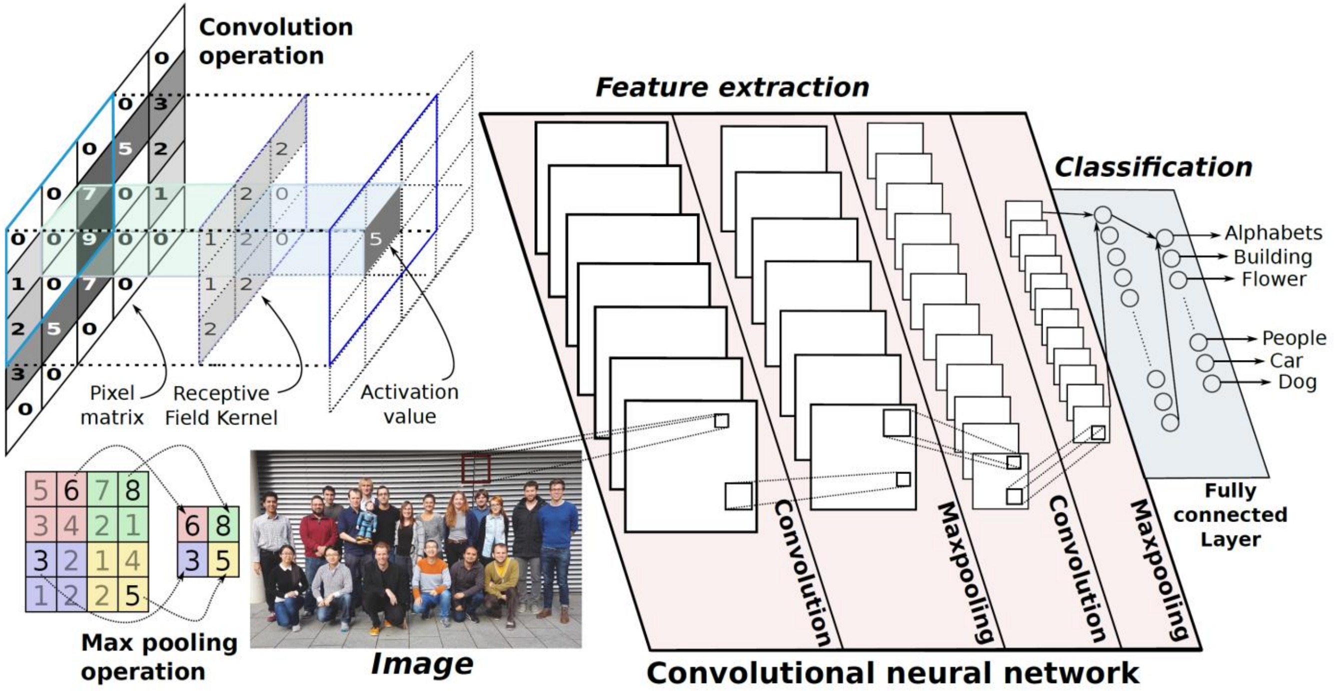
Feature Representation Learning

- **Learning features directly from data**
  - Conventional features are costly and labour intensive
    - Hand-crafted features may not capture suitable discriminative information for task at hand
  - Solution
    - Create learning algorithms that extract their own features
  - Widely used in conjunction with deep learning
    - Convolutional Neural Networks
      - Multiple layers of filtering

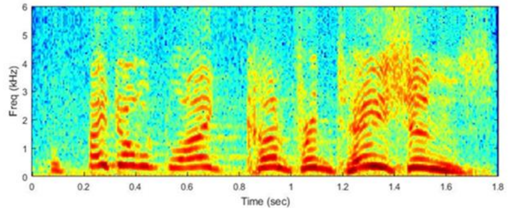


Universität Augsburg University

- **Convolutional Neural Network (CNN)**
  - Special form of feed-forward network
    - Reuse the same neurons for repetitive convolution tasks
    - Convolutional kernels recognise features in signal



Feature learning with CNNs



can repeat many times

Convolution

- Sets of filters used to identify patterns within the signal
    - Network learns weights associated with each filter
- Similar patterns appear in mutiple regions
    - Learns importance of different features to the task at hand

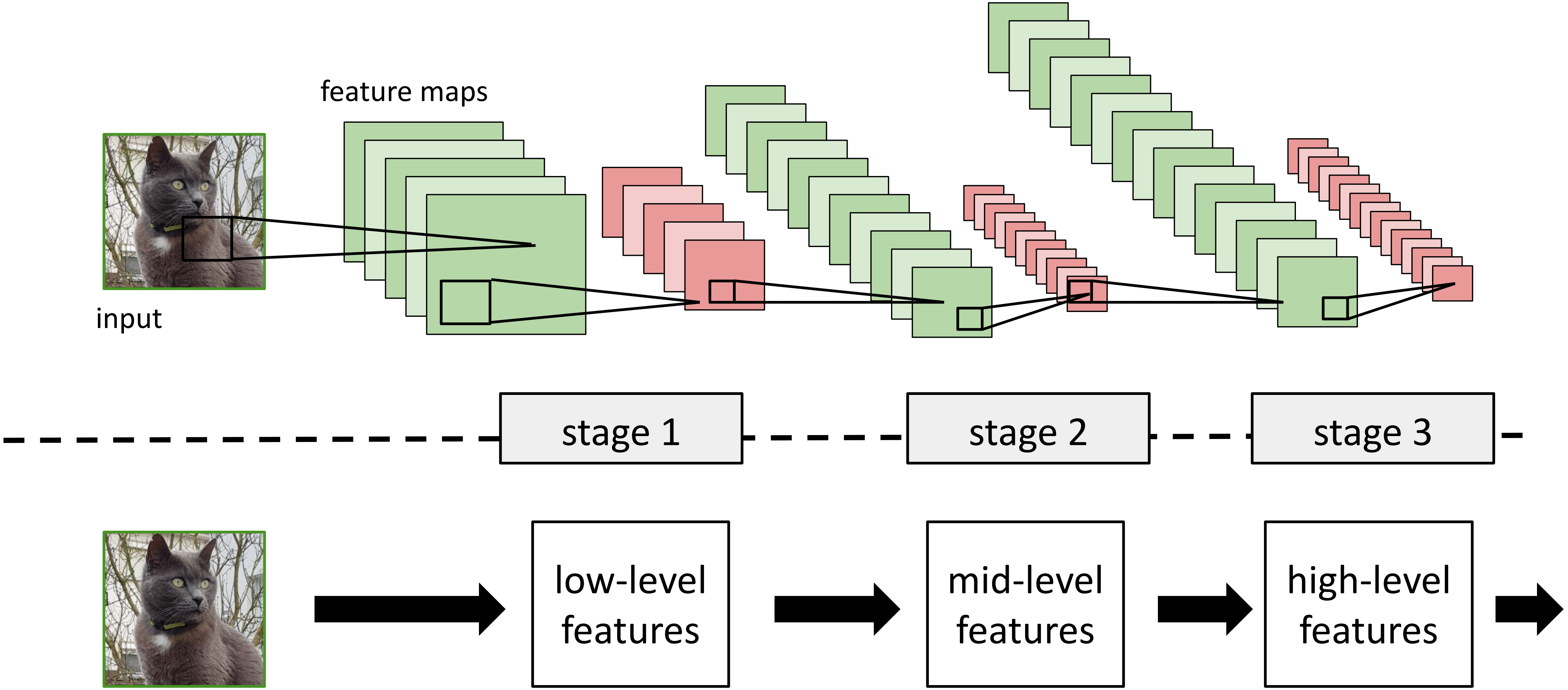
Max Pooling

- Down sampling operation to reduce information
- Learnt features invariant to small translations (shifts) in the input

Convolution

Max Pooling → Flatten → Classifier → prediction

Representation Learning with CNNs

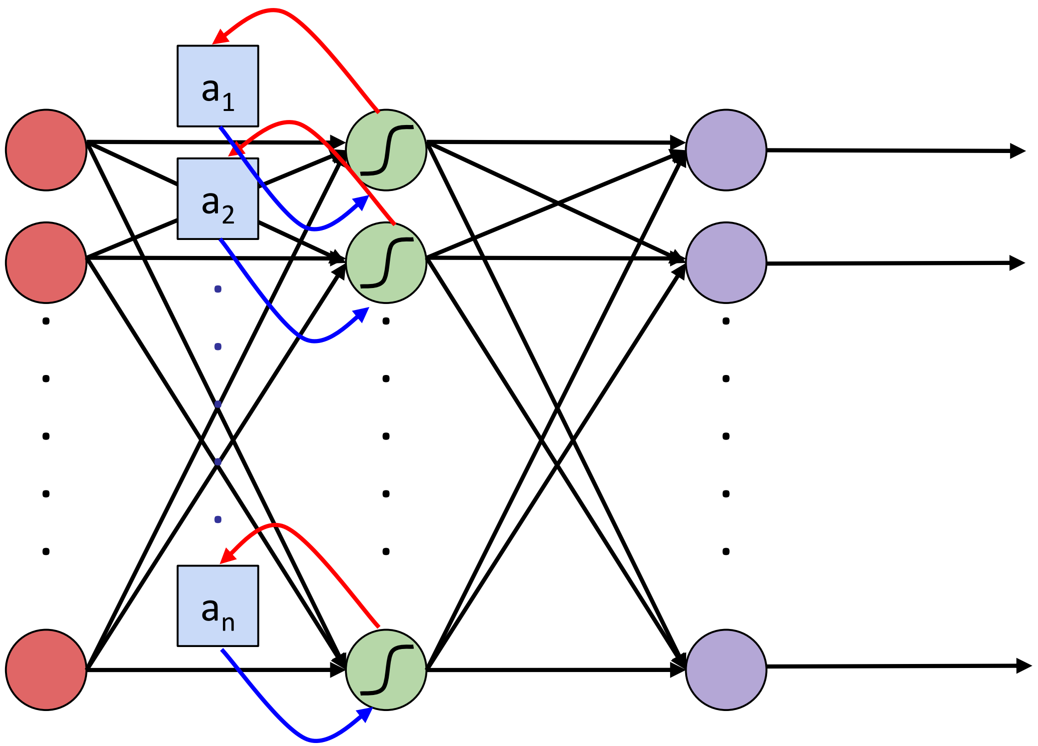


Recurrent Neural Network

- Inclusion of feedback into the network structure

Output of hidden layer are **stored** in the memory

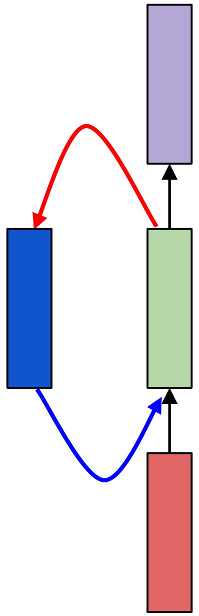
Values in the memory are considered as **additional input** in the next time step



Recurrent Neural Network

- Inclusion of feedback into the network structure

memory to store the hidden states



Process a sequence of vectors $\boldsymbol{x}$ by applying a *recurrence formula* at every time step:

$$h_t = f_W(h_{t-1}, x_t)$$

current state

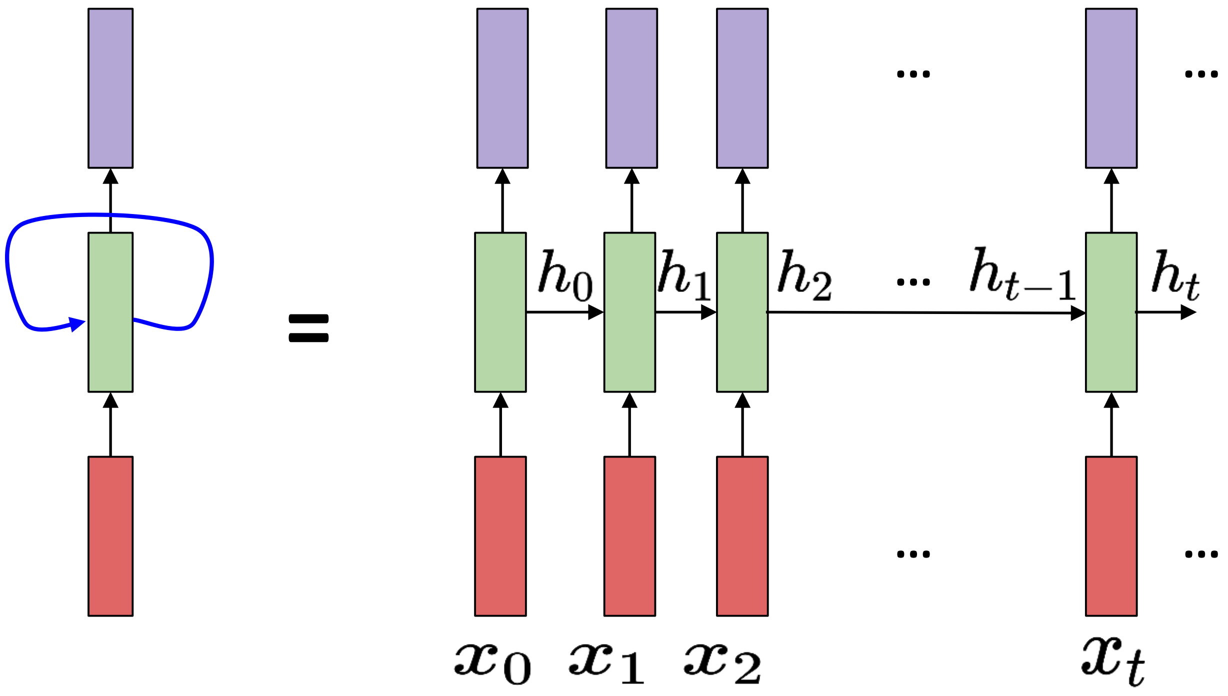
function with parameters W

state in memory (previous state)

current input vector

The same function and the same parameters are used at every time step.

Unrolled RNN



- Reuse the same weight matrix at every time step
- Makes the network easier to train

Recurrent Neural Network

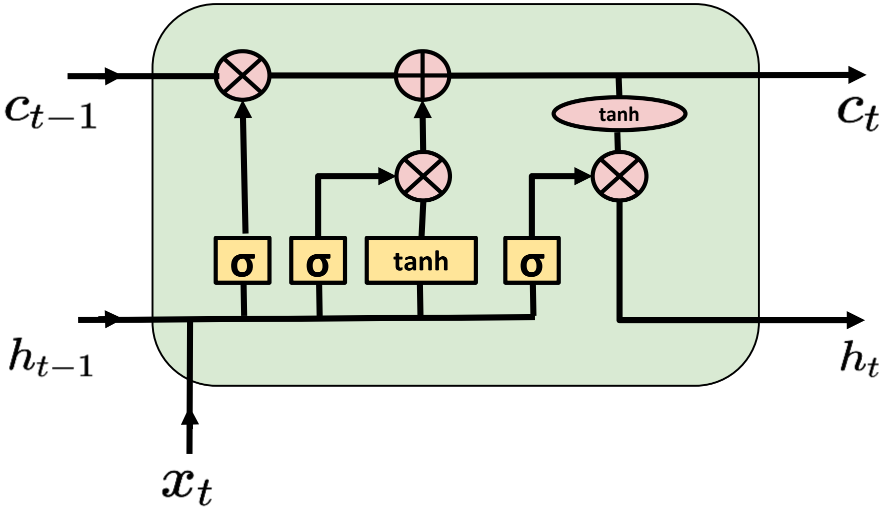
- Susceptible to vanishing gradient problem during training

Solution: Gated Networks

- Gated RNNs are based on the idea of creating paths that have derivatives that neither vanish nor explode
- Gated RNNs have connection weights that may change at each time step
- Gated RNNs also allow a network to forget an old state
- Instead of manually deciding when to clear the state, the network to learn to decide when to do it.

Image source: https://colah.github.io/

- **Long-Short Term Memory**
- There are four interacting networks: cell state, input gate, forget gate, output gate



$$g_t = tanh\,(W_g \cdot [h_{t-1}, x_t] + b_g)$$

$$i_t = \sigma\,(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$f_t = \sigma\,(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

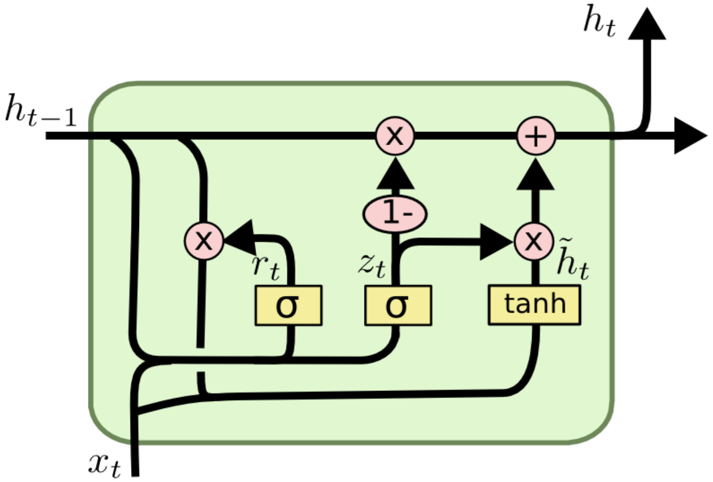
$$o_t = \sigma\,(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot tanh(c_t)$$

Image source: https://colah.github.io/

Gated Recurrent Unit

- Single gating unit simultaneously controls the forgetting factor and the decision to update the state unit



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t] + b_z\right)$$

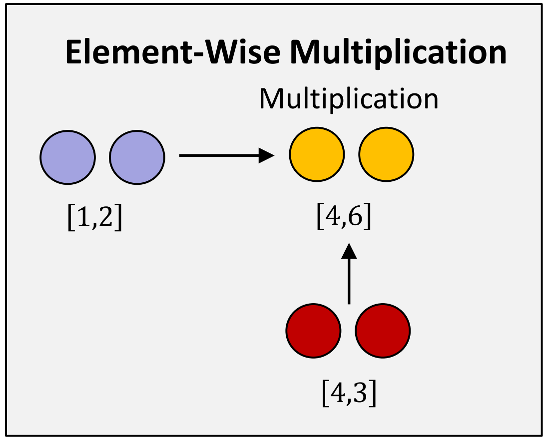
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t] + b_r\right)$$

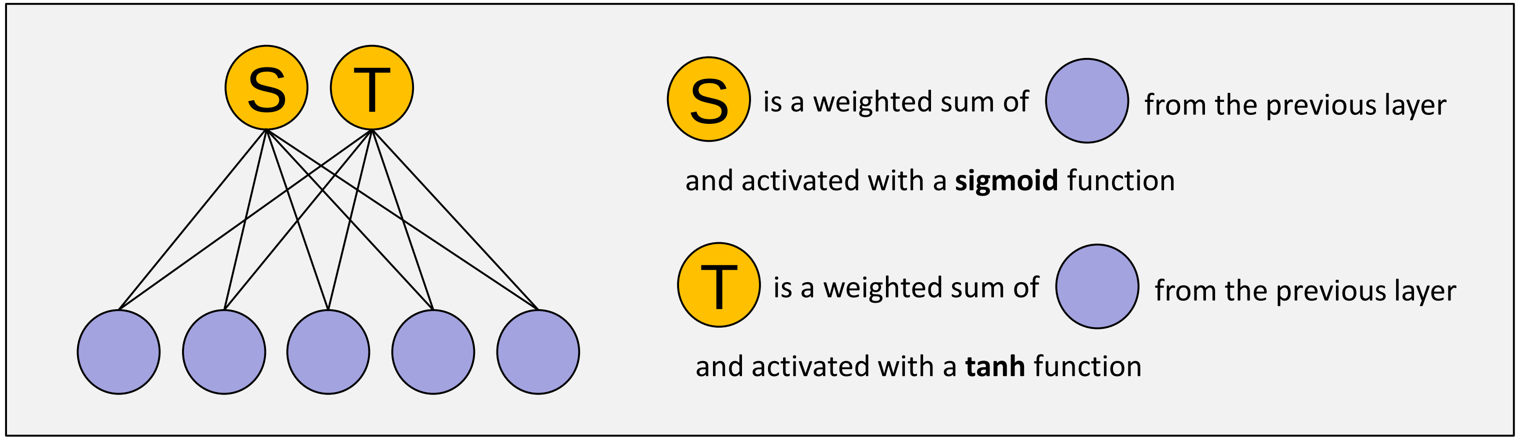
$$g_t = tanh\left(W_g \cdot [r_t \odot h_{t-1}, x_t] + b_g\right)$$

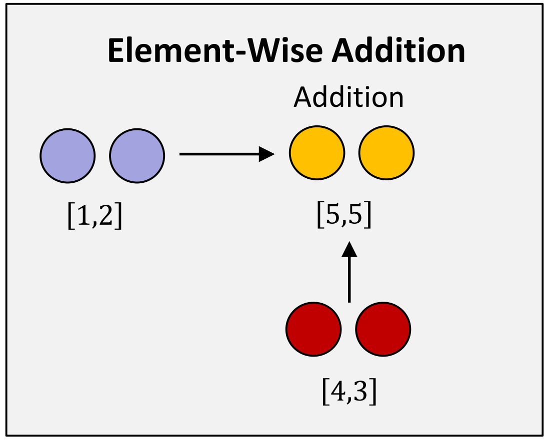
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot g_t$$

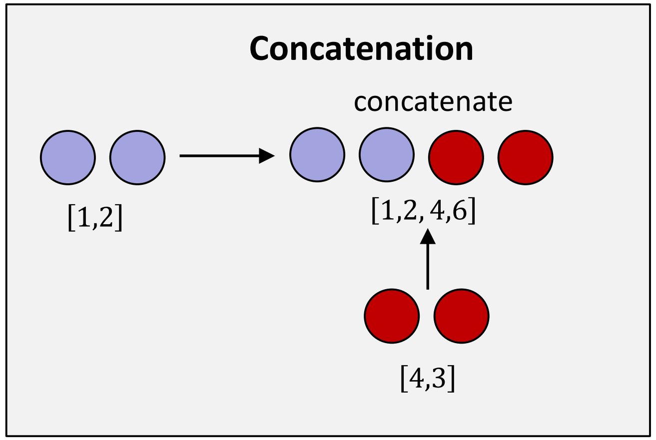
Universität Augsburg University

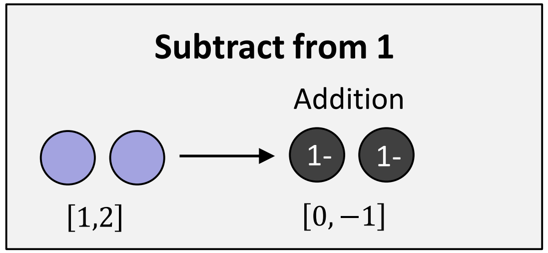
- Guide to upcoming illustrations:



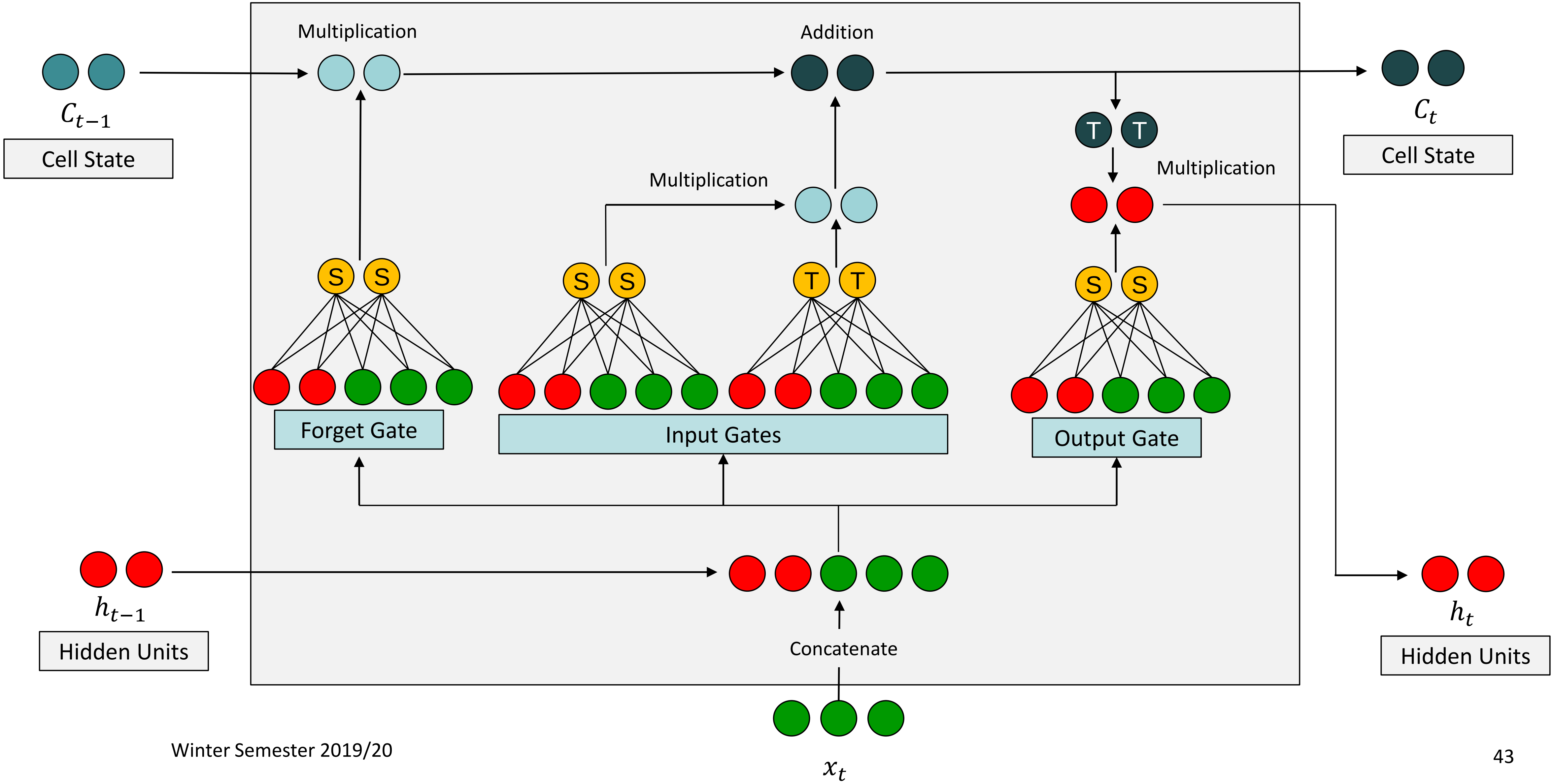




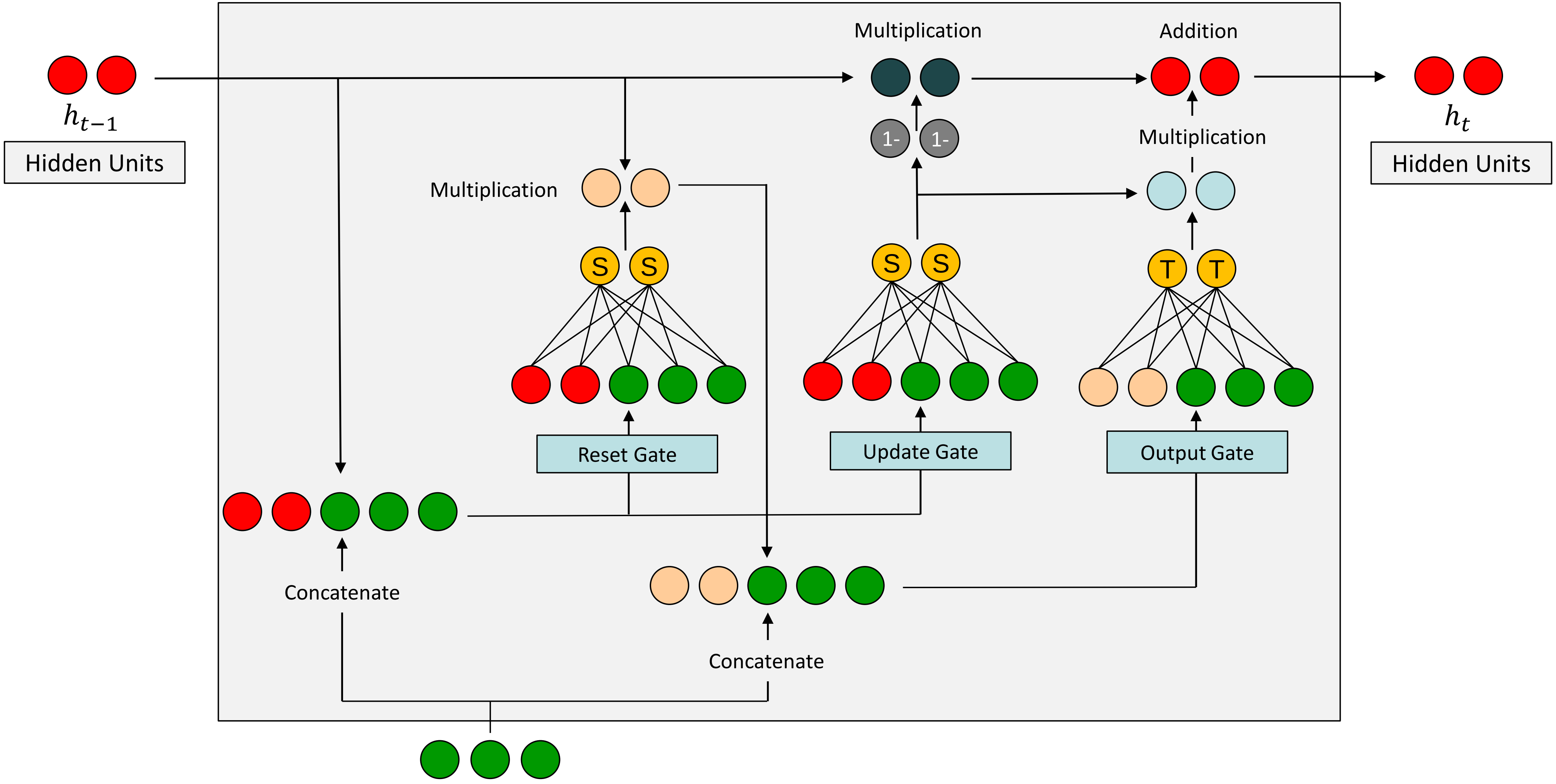




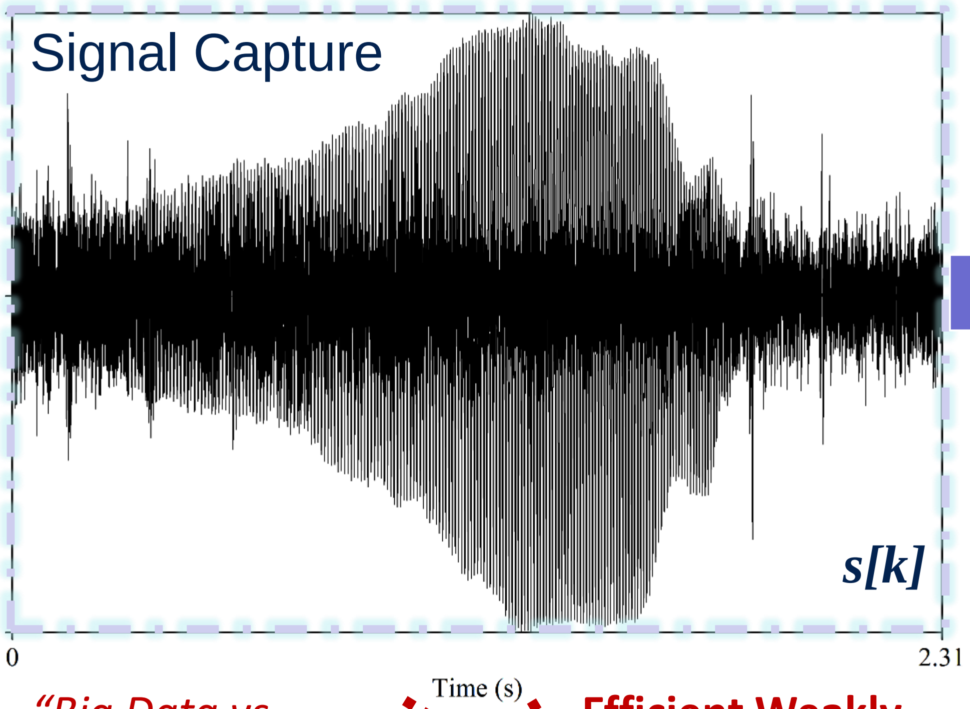
Long Short-term Memory (LSTM)



Gated Reccurent Unit



End-2-End Learning



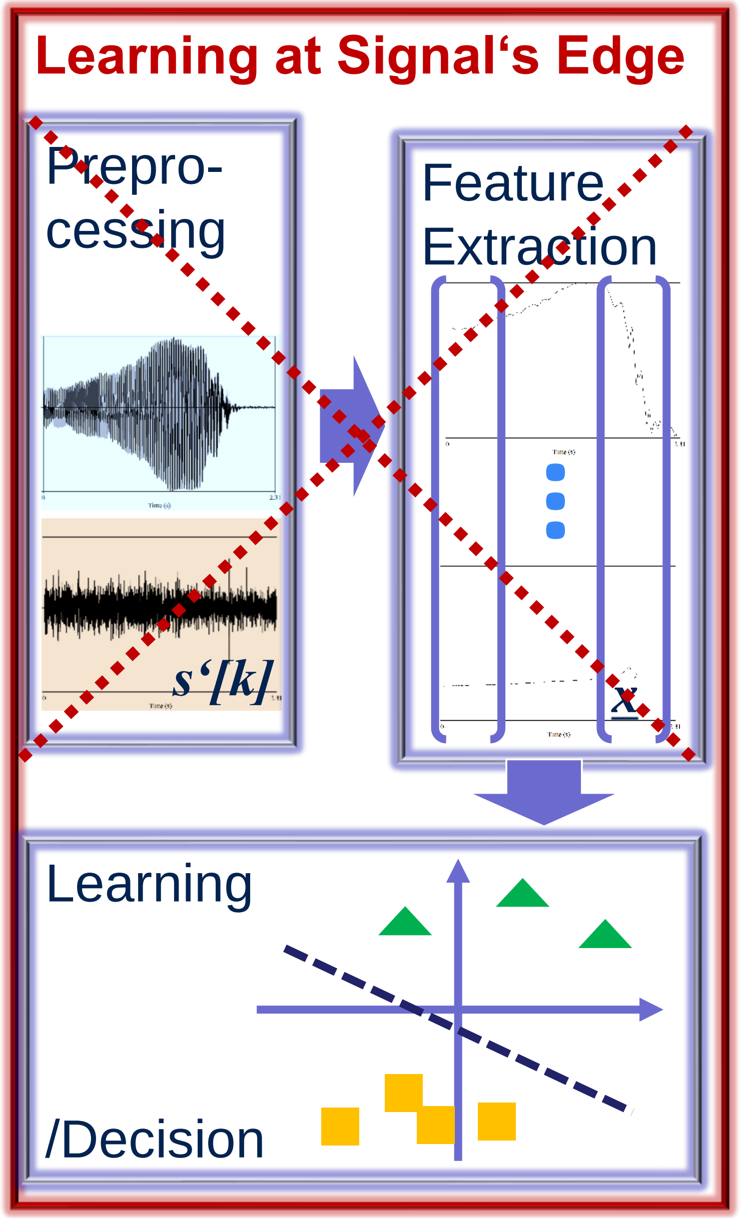
"Big Data vs Little Labels"

Efficient Weakly Supervised

<label>

(e.g.. "burst")

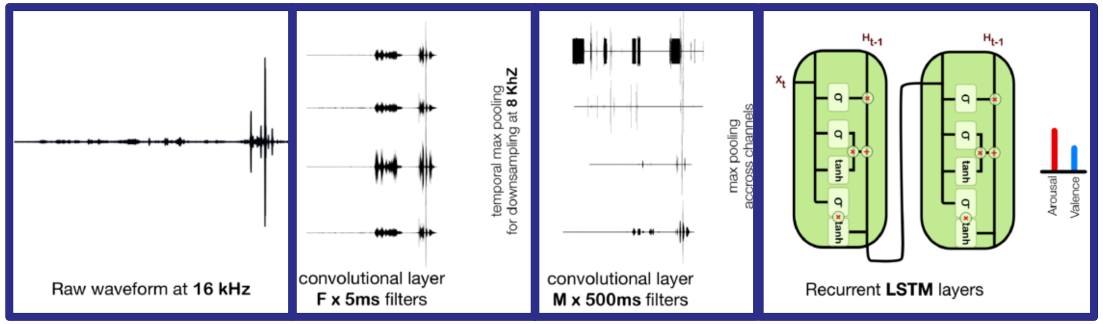
y

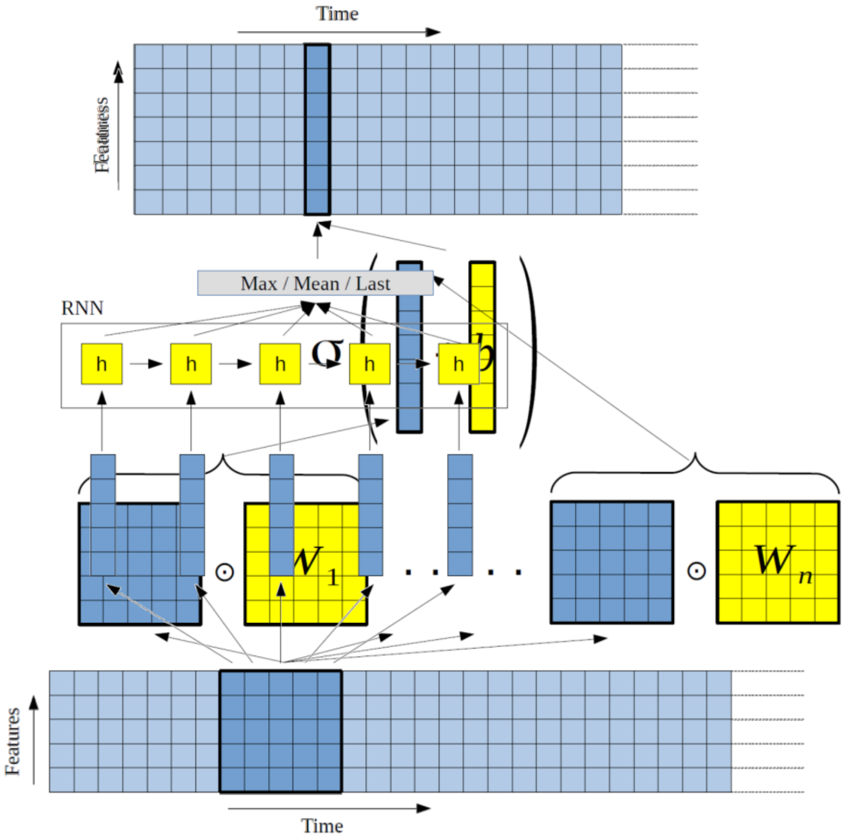


Pattern Recognition 2.0?

- CNN + LSTM → CLSTM ?

Arousal	CCC
Baseline	.366
e2e	.686



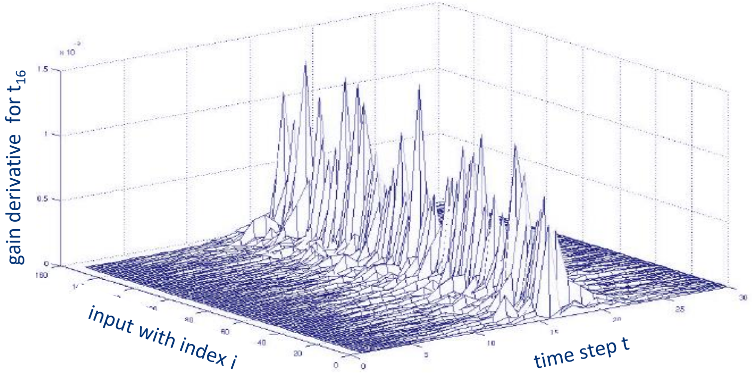


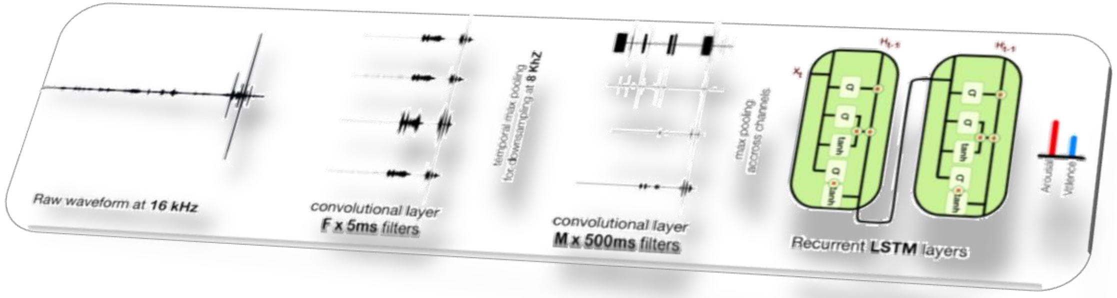
G. Trigeorgis, F. Ringeval, R. Bruckner, E. Marchi, M. Nicolaou, B. Schuller, and S. Zafeiriou, “Adieu Features? End-to-End Speech Emotion Recognition using a Deep Convolutional Recurrent Network,” in Proceedings 41st IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2016, (Shanghai, P. R. China), pp. 5200–5204, IEEE, IEEE, March 2016.

Universität Augsburg University

End-to-End – a black box?

- CNN activations correlate with standard speech features

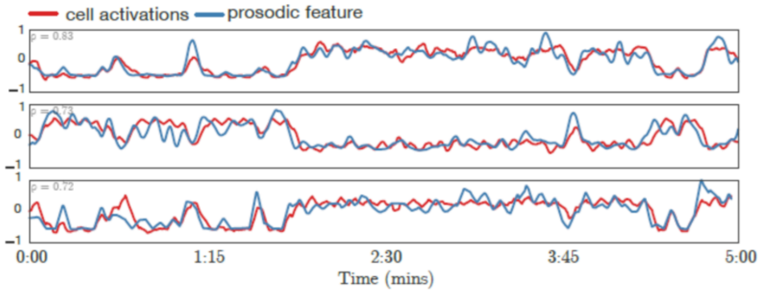




energy range (.77)

loudness (.73)

F0 mean (.71)



Welcome!

What is it?

Who are we?

Deep Learning

What will we do?

Why?

What do we expect?

Knowledge

- The lecture content includes an overview of Neural Network developments, covering all aspects from basic concepts to complex models. The course will cover Neural Network architectures that are suitable for a variety of data types and signals from different domains, such as audio, speech, vision, and text

Skills

- In the Tutorials, students will be familiarised with the latest Deep Learning toolkits such as Tensorflow and Keras and learn how to train and evaluate Deep Neural Networks for different applications.

Rest of the course

Lectures: Tuesdays 12:15 – 13:45 in 2045 N

- Topics Covered:
  - Machine Learning
  - Feed Forward, Convolutional & Recurrent Networks
  - Attention Modelling and Connectionist Temporal Classification
  - Natural Language Processing
  - Explainability

Tutorials: Wednesdays 12:15 - 13:45 in 1058 N

- Skills gained in using
  - Python, Tensorflow, Keras

Lecture Timetable

Date	Topic
15.10.2019	Introduction
22.10.2019	Tutorial: Introduction to Python and Numpy
29.10.2019	Gradient Descent
5.11.2019	Feed Forward Networks
12.11.2019	Recurrent Neural Networks
19.11.2019	Convolutional Neural Networks
26.11.2019	Regularisation in Neural Networks
03.12.2019	Sequence to Sequence Learning
10.12.2019	Introduction to Natural Language Processing
17.12.2019	Lecture by Prof. Schuller
Christmas / New Year Break	
07.01.2019	Data Representation learning
14.01.2019	Reinforcement Learning
21.01.2019	Next Generation Neural Networks
28.01.2019	Explainable AI
04.02.2019	Wrap-Up Lecture

Tutorial Timetable

Date	Topic
16.10.2019	No Tutorial
23.10.2019	Lecture: Machine Learning Concepts
30.10.2019	Maths for Deep Learning
6.11.2019	Forward Propagation
13.11.2019	Gradient Descent & Back Propagation
20.11.2019	Recurrent Neural Networks
27.11.2019	Convolutional Neural Networks
04.12.2019	Convolutional Neural Networks for Audio
11.12.2019	Autoencoders
18.12.2019	Sequence to Sequence Learning
	Christmas / New Year Break
08.01.2019	Generative Adversarial Networks
15.01.2019	
22.01.2019	Group Challenge
29.01.2019	
05.02.2019	Group Challenge Presentation

Welcome!

What is it?

Who are we?

Deep Learning

What will we do?

Why?

What do we expect?

What do we expect?

Basic knowledge

- In signal processing and additional knowledge in the fields of Machine Learning, Data Mining, or more generally Pattern Recognition is of advantage.

Target groups

- Include post-graduate students of Electronics and Telecommunications Engineering, Information Technology, Computer Sciences and related studies.

The teaching language is English.

Interest & Interaction!

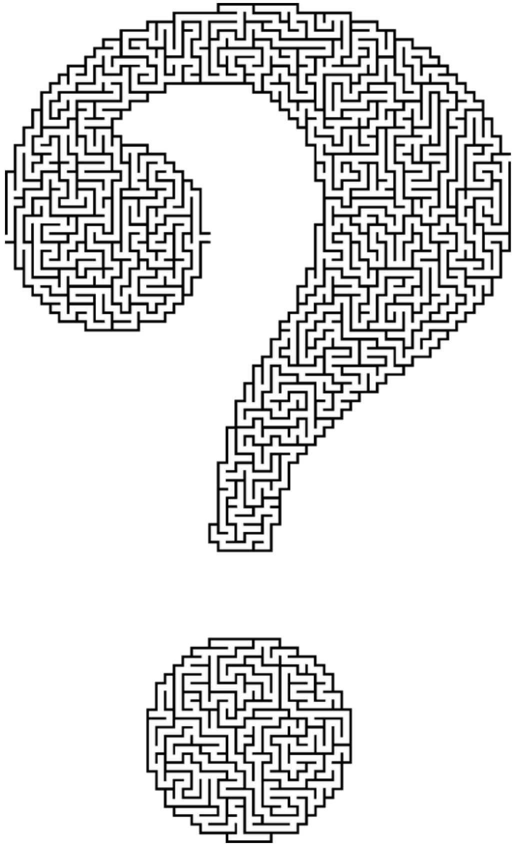


Image Source: https://pixabay.com

Tutorials

- Start: 22nd Oct
- Presentation of Background to Current Topic
- 10 Weekly Exercise Sheets + 1 Three-Week Group Challenge
- Team Work encouraged (Starting from Ex. 3)
- Programming, Maths Exercises, Additional Questions
- Upload by Monday on Digicampus
- Submission by Monday Evening (Next Week) via Email
- Review of Exercises Wednesday after Submission

Tutorials: Bonus Point System

- Solve Exercise Sheets and Participate in Tutorial
- Average of ~20 Possible Points per Exercise Sheet
- Additional 50% of Points for Presentation in Tutorial
- 75% of Points needed for Exam Bonus
- Exam Bonus: 0,3/0,4
  - E.g., 1,3 → 1,0/ 1,7 → 1,3
- Bonus does **NOT** help you Pass the Exam
  - 4,3 still fails, even if you have Bonus Points

Written Exam

- Combination of multiple choice and written answers
- Need to get 50% to pass the course
- Exam language will be English
- Full details, including dates, given later in the course
- Practise questions will be given in lectures and tutorials

Introductory Lecture

Welcome!

What is it?

Who are we?

Deep Learning

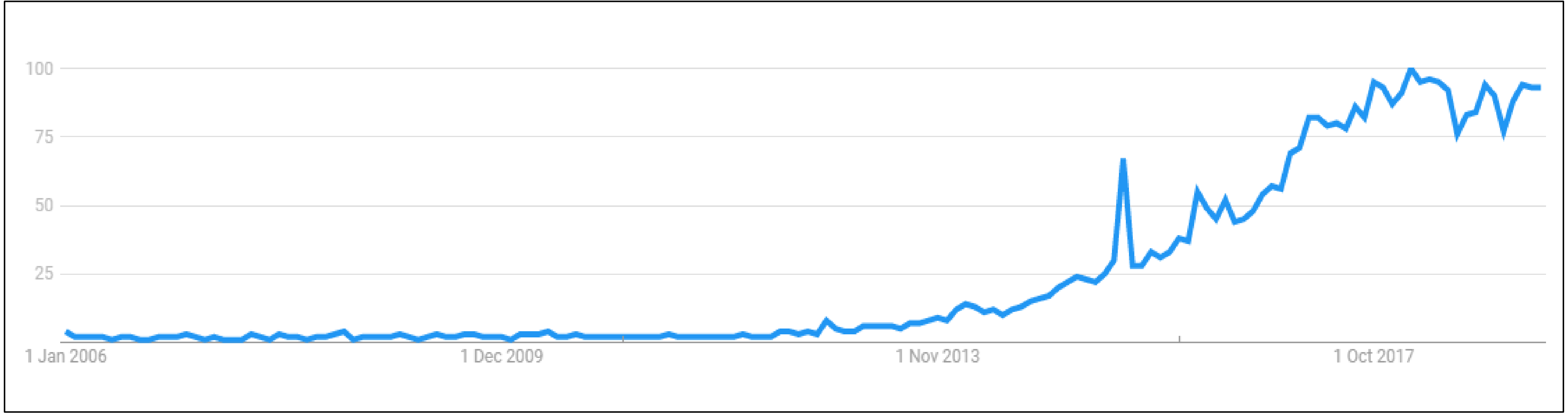
What will we do?

Why?

What do we expect?

Growing Interest

- Google trends graph on the term ‘Deep Learning’
  - Clear trend in increasing world-wide interest



Why study Deep Learning?

Growing Job Market

- Germany seeking to become a global AI powerhouse
  - 3 Billion Euro to be invested by 2025
    - Mobility, Health, Energy, Industry 4.0
  - 100 new AI chairs to be created
  - 12 new centres for research
  - Attractive working conditions & salaries

Video source: https://www.ki-strategie-deutschland.de

Universität Augsburg University

Money

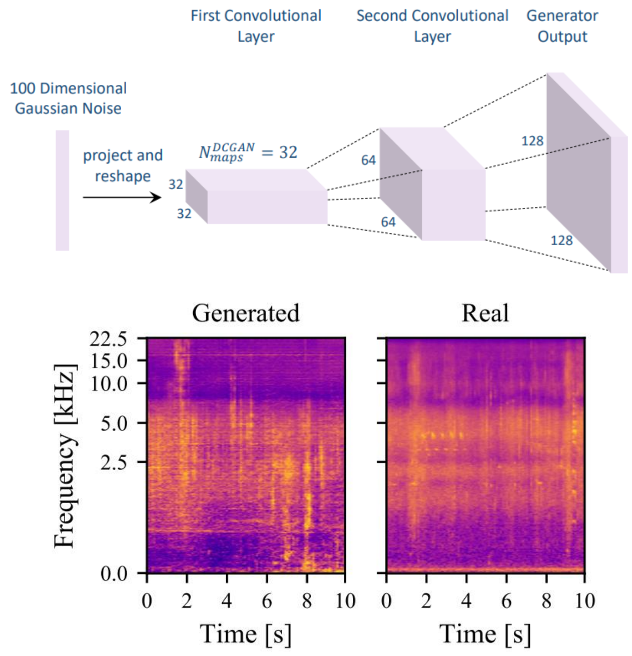
- AI experts can command huge salaries
  - AI specialist can make between $300,000 and $500,000 a year in salary and stock
    - E.g. Average wage at Google Deep Mind ~ $345,000
    - E.g. OpenAI employed 52 people in 2016 and spent more than $7 million on salaries
      - Source: New York Times


The New York Times
Tech Giants Are Paying Huge Salaries for Scarce A.I. Talent


The New York Times
A.I. Researchers Are Making More Than $1 Million, Even at a Nonprofit

Interesting Research

- **Generative Adversarial Networks (GANs)**
  - Deep neural net architectures comprised of two networks, pitting against each other in a zero-sum game
  - Huge Potential
    - Learn to mimic the distribution of data
  - Used to create and augment data sources

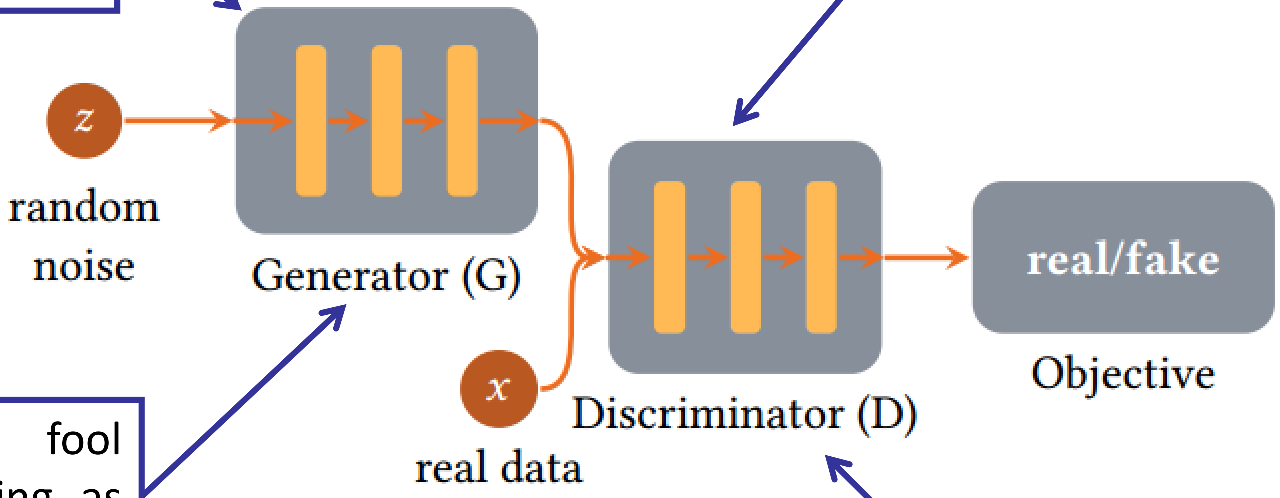


S. Amiriparian, et al, "A Fusion of Deep Convolutional Generative Adversarial Networks and Sequence to Sequence Autoencoders for Acoustic Scene Classification", EUSIPCO 2018

Generative Adversarial Networks

Maps noise sampled from a pre-defined distribution onto a training data distribution

Predict if a given sample is real or generated

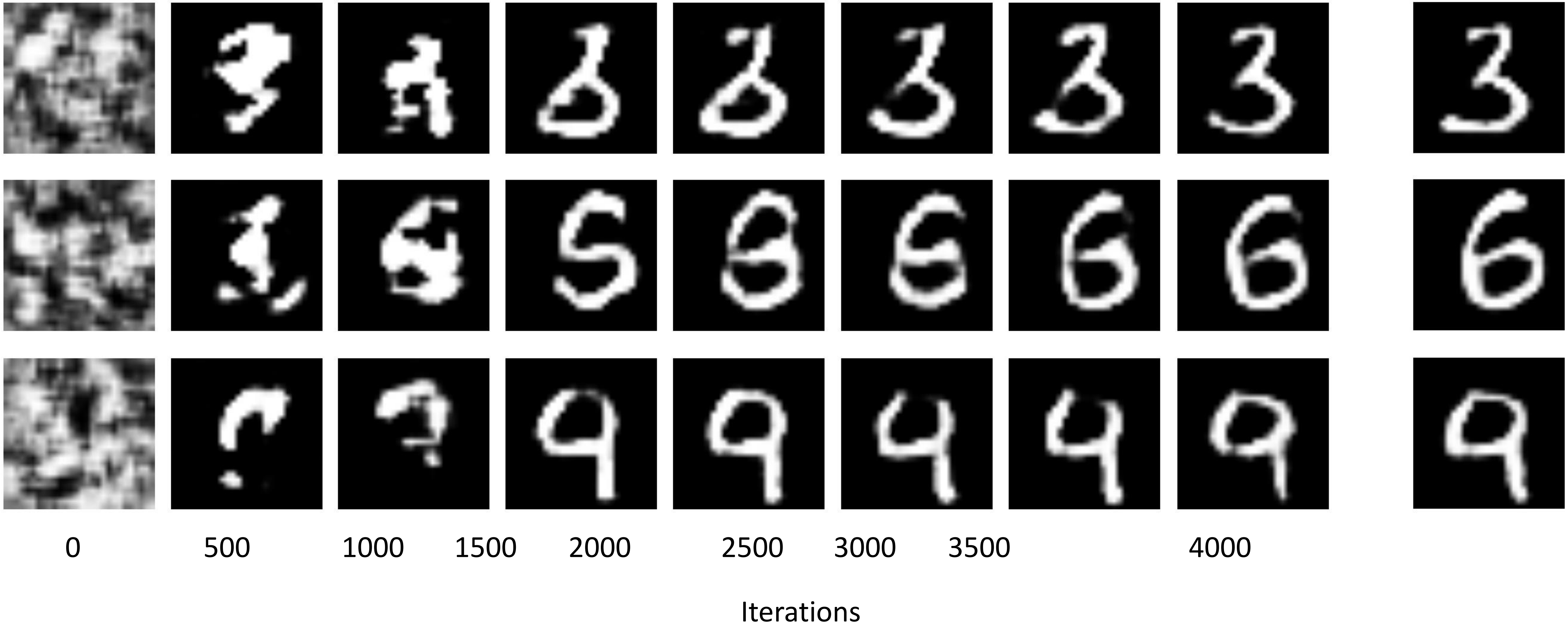


Game ends when discriminator cannot pick the difference between artificial and real data

Generator learning to fool discriminator by producing as realistic samples as possible

Discriminator learns to differentiate between real and fake samples

GAN Example:



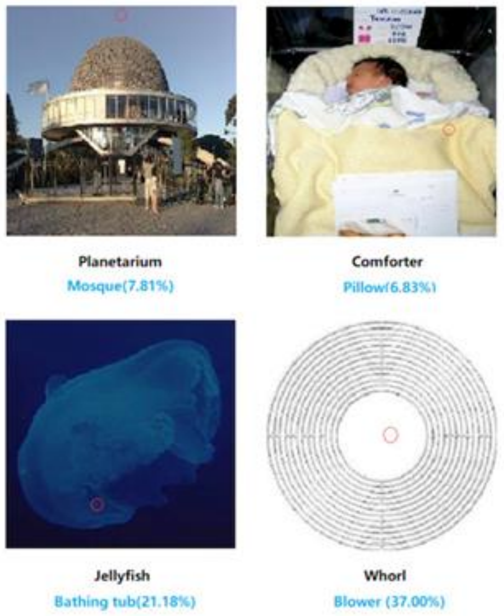
Plenty of Research Still needed

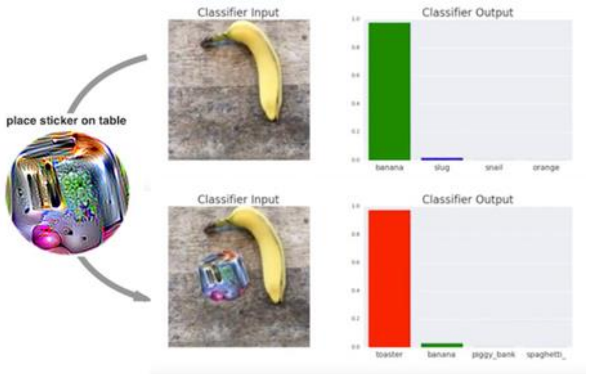
- Prominent AI failures
  - **UBER, 2018**: An experimental autonomous Uber vehicle struck and killed a pedestrian in Tempe, Arizona
  - **Goldman Sachs, 2018**: AI predictions the 2018 World Cup were almost all wrong
  - **Amazon 2017:** AI-enabled recruiting software is found to be gender biased
  - **Volvo 2017:** Self-driving technology struggles to identify kangaroos in the road
  - **IBM, 2016:** Watson for Oncology cancel after spending $62 million on the project after system was found to give unsafe advice
  - **Microsoft, 2016:** Twitter chatbot began stating 9/11 conspiracy theories and Nazi sentiments
  - **Google 2015**: Image recognition algorithms in Google Photos was found to be labelling black people as gorillas

Why study Deep Learning?

Plenty of Research Still needed

- **Data Requirements**
  - Robust deep learning solutions require massive amounts of training data
- **Privacy Concerns**
  - Large deep neural networks are difficult to implement in edge computing scenarios
- **Butterfly effects / Adversarial attacks**
  - Small variations in the input data can lead to drastically different results.
  - By adding small amounts of noise it is possible to fool deep learning networks
- **Increasing system trustability**
  - Cope with inherent biases in training data





Where to next?

- **Increased Explainability**
  - AI based systems are achieving remarkable results
  - These results are often achieved using `black-boxes'
    - Data is fed in and a predictive output generated
    - The system does not provide any information concerning how it arrived at the predicted value.
  - This issue is particularly pronounced in deep learning
    - Specific deep learning system may have nodes and connections numbering in the millions
    - Internal operations which should be considered highly declarative.

Explainable AI

- **Local Interpretable Model-Agnostic Explanations (LIME)**
  - Trains local surrogate models to approximate the predictions of the underlying black box model
  - **Key Steps**
    - Trains your (black-box) model
    - Select instance to explain
    - Create perturbed dataset
    - Train a weighted, interpretable model, on perturbed dataset variations
    - Explain the prediction using model

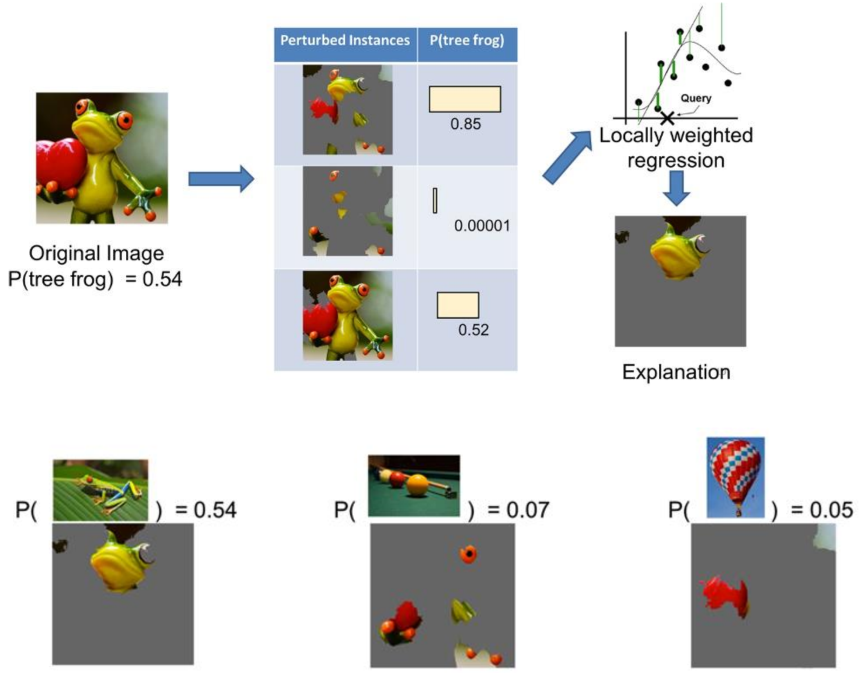


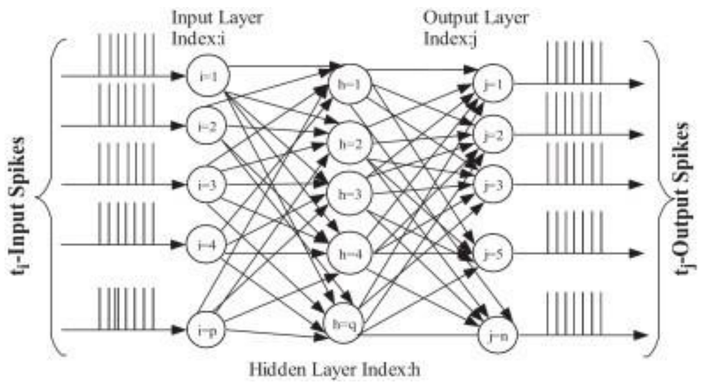
Image Source: https://distill.pub/2017/feature-visualization/

- **Feature Visualisation**
  - DNN's learn features in their hidden layers
  - Visualisation of these features are possible
    - Finding the input that maximizes the activation of a unit
  - **Key steps in Feature Visualisation**
    - Start from random noise
    - Place constraints on the update
      - Ensure that only small changes are allowed
    - Apply steps to reduce noise in updates
      - Jittering, rotation or scaling to the image

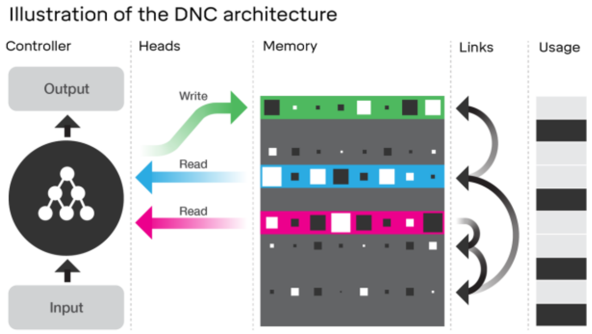
Objects (layers mixed4d & mixed4e)

Plenty of Research Still needed

- Next Generation Networks:
  - **Spiking Neural Networks**
    - Data driven event based computation
    - Encode information as events in time
  - **Differentiable Neural Computer**
    - Neural network with a memory matrix from which it can read and write data
    - Allows network to have long-term memory
  - **Meta-Learning**
    - Networks trained to be able to learn the learn processes



Source: https://towardsdatascience.com/



Source: https://becominghuman.ai