Deep Learning

Knowledge Discovery and Data Mining 2 (VU) (706.715)

Roman Kern

ISDS, TU Graz

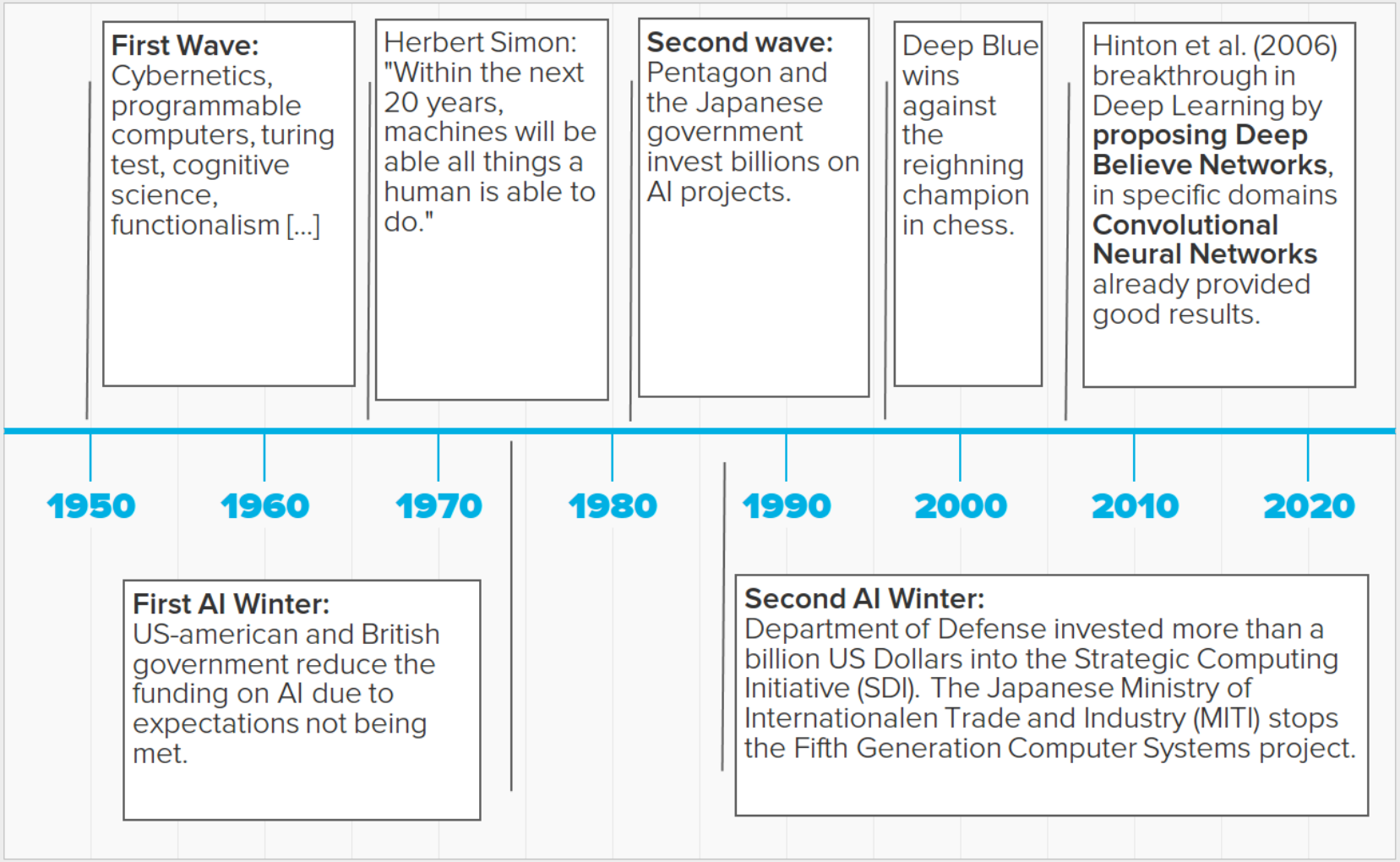
2018-05-24

Outline

Introduction to Deep Learning

What & Why

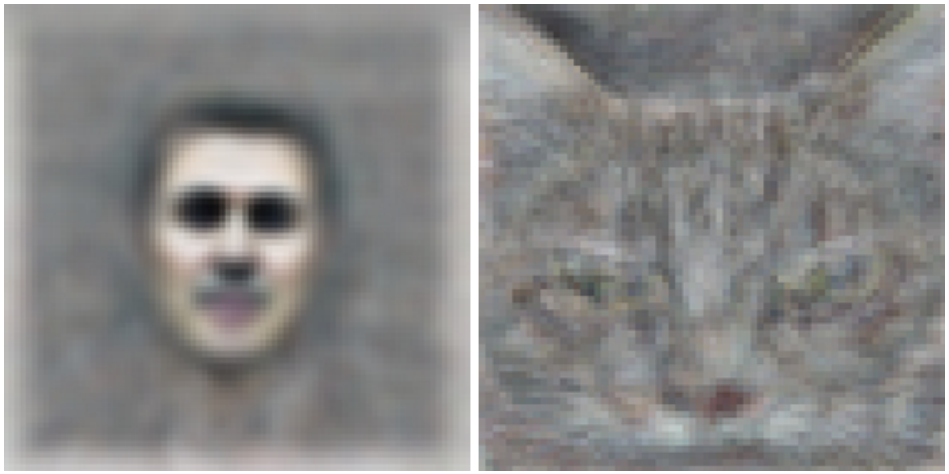
History of Artificial Intelligence



Success Stories of Deep Learning

Unsupervised high-level feature learning

- Using a deep network of 1 billion parameters, 10 million images (sampled from YouTube), 1000 machines (16,000 cores) x 1 week.
- Evaluation
  - ImageNet data set (20,000 categories)
  - 0.005% random guessing
  - 9.5% state-of-the-art
  - 16.1% for deep architecture
  - 19.2% including pre-training



https://research.google.com/archive/unsupervised_icml2012.html

Success Stories of Deep Learning

- Primarily on speech recognition and images
- Interest by the big players
- Facebook
  - Face recognition
  - `https://research.facebook.com/publications/480567225376225/deepface-closing-the-gap-to-human-level-performance-in-face-veri`
- Baidu
  - Speech recognition
  - `https://gigaom.com/2014/12/18/baidu-claims-deep-learning-breakthrough-with-deep-speech/`
- Microsoft
  - Deep learning technology centre
  - e.g. NLP - Deep Semantic Similarity Model
  - `http://research.microsoft.com/en-us/projects/dssm/`

Prerequisite Knowledge

- Neural Networks
  - Backpropagation
  - Recurrent neural network (good for time series, NLP)
- Optimization
  - Generalisation (over-fitting), regularisation, early stopping
  - Logistic sigmoid, (stochastic) gradient descent
- Hyper Parameters
  - Number of layers, size of e.g. mini-batches, learning rate, ...
  - Grid search, manual search, a.k.a Graduate Student Descent (GSD)

Neural Network Properties

- 1-layer networks can only separate linear problems (hyperplane)

Neural Network Properties

- 1-layer networks can only separate linear problems (hyperplane)
- 2-layer networks with a non-linear activation function can express any continuous function (with an arbitrarily large number of hidden neurons)

Neural Network Properties

- 1-layer networks can only separate linear problems (hyperplane)
- 2-layer networks with a non-linear activation function can express any continuous function (with an arbitrarily large number of hidden neurons)
- For more than 2 layers, one needs fewer nodes $\rightarrow$ therefore one wants have deep neuronal networks

Neural Network Properties

- Back propagation does not work well for more than 2 layers
  - Non-convex optimization function
  - Uses only local gradient information
  - Depends on initialisation
  - Gets trapped in local minima
  - Generalisation is poor
  - Cumulative backpropagation error signals either shrink rapidly or grow out of bounds (exponentially) (Hochreiter, 1991)
- Severity increases with the number of layers
- Focus shifted to convex optimization problems (e.g., SVM)

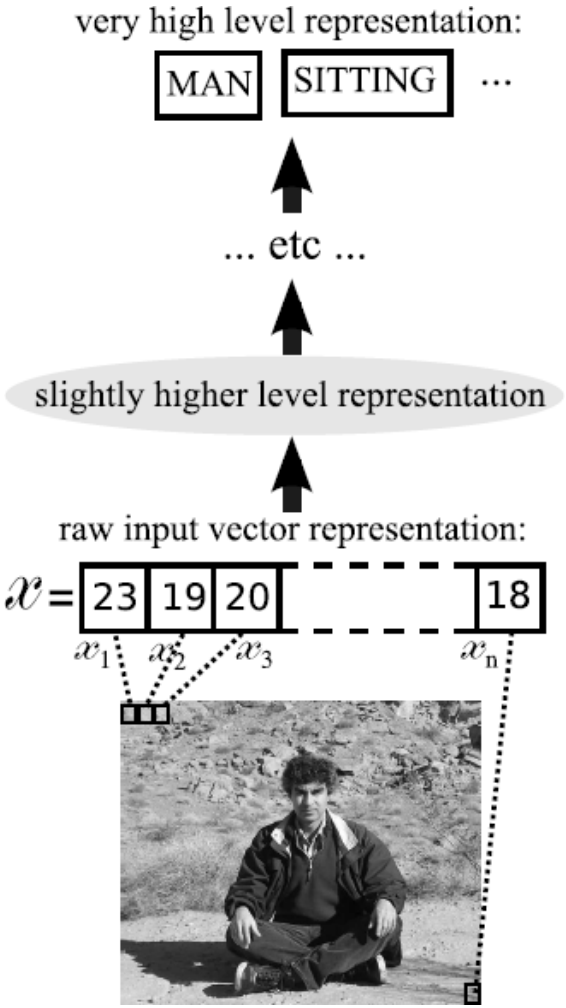
Deep Learning Approaches

Overview of the most common techniques

Definition of Deep Learning

- Several definitions exist
- **Two key aspects:**
  1. models consisting of **multiple** layers or stages of **nonlinear** information processing
  2. methods for **supervised or unsupervised learning** of feature representations at successively higher, more abstract layers
- Deep Learning architectures originated from, but are not limited to artificial neural networks
- Contrasted by conventional **shallow learning** approaches
- Not to be confused with deep learning in educational psychology:
  - "Deep learning describes an approach to learning that is characterized by active engagement, intrinsic motivation, and a personal search for meaning."

Example



objects

edges, shapes, etc.

raw pixel values

Y. Bengio (2009). Learning Deep Architectures for AI. Foundations and Trends in Machine Learning, 2(1), 1–127.

Deep Learning vs. Shallow Learning

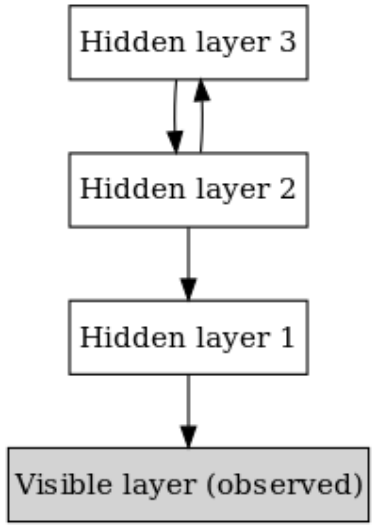
- When does **shallow learning** end and **deep learning** begin?
- What is the depth of an machine learning algorithm?
- **Credit assignment path (CAP)**: chain of causal links between input and output
- **Depth**: length of CAP starting at the first **modifiable** link
- Examples:
  - Feed-forward network: depth = number of layers
  - Network with fixed random weights: depth = 0
  - Network where only the output weights are trained (e.g., Echo State Network): depth = 1
  - Recurrent neural network: depth = length of input (potentially unlimited)
- **Deep Learning:** depth > 2; **Very Deep Learning:** depth > 10

History

- The concept of deep learning originated from artificial neural network research
- The deep architecture of the human brain is a major inspiration: it successfully incorporates learning and information processing on multiple layers
- However, training ANNs with more than two hidden layers yielded poor results
- Breakthrough 2006 (Hinton et al.): **Deep Belief Networks** (DBN)
- Principle: training of intermediate representation levels using **unsupervised learning**, which can be performed **locally at each level**

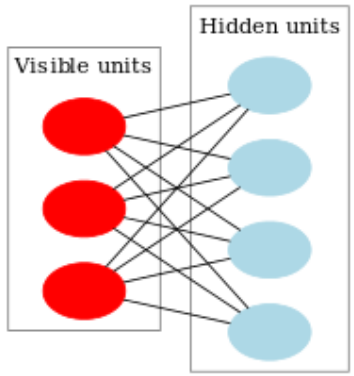
Deep Belief Network (DBN)

- A probabilistic, generative model composed of **multiple simple learning modules** that make up each layer
- Typically, these learning modules are **Restricted Boltzmann Machines** (RBMs)
- The top two layers have symmetric connections between them. The lower layers receive top-down connections from the layer above
- **Greedy layer-wise training**: Each layer is successively trained on the output of the previous layer
- Can be used for **pre-training** a network followed by fine tuning via backpropagation



Restricted Boltzmann Machine (RBM)

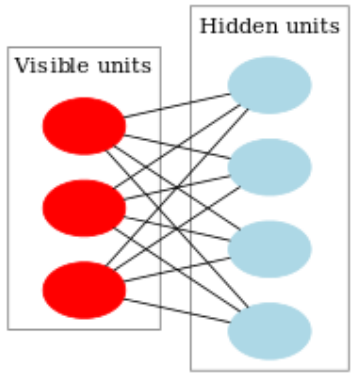
- Stochastic artificial neural network forming a **bipartite graph**
- The network learns a **representation** of the training data presented to the visible units
- The hidden units model statistical dependencies between the visible units
- Try to optimise the weights so that the likelihood of the data is maximised



$$P(h_j = 1|v) = \sigma\left(b_j + \sum_i v_i w_{ij}\right)$$
$$P(v_i = 1|h) = \sigma\left(a_i + \sum_j h_j w_{ij}\right)$$
$$\sigma(z) = 1/(1 + e^{-z})$$

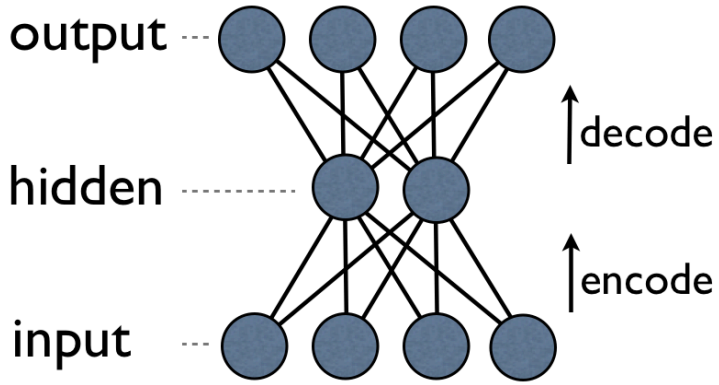
Restricted Boltzmann Machine (RBM)

- Activations within one layer are **conditionally independent** given the activations in the other layer
- $\rightarrow$ efficient training algorithm (**Contrastive Divergence**):
  1. From a training sample $\mathbf{v}$, compute the probabilities of the hidden units and sample a hidden activation vector $\mathbf{h}$ ($\mathbf{v}\mathbf{h}^T$ ... positive gradient).
  2. From $\mathbf{h}$, sample a reconstruction $\mathbf{v}'$ of the visible units, then resample the hidden activations $\mathbf{h}'$ from this (Gibbs sampling; $\mathbf{v}'\mathbf{h}'^T$ ... negative gradient).
  3. Update the weights with $\Delta w_{i,j} = \epsilon(\mathbf{v}\mathbf{h}^T - \mathbf{v}'\mathbf{h}'^T)$.



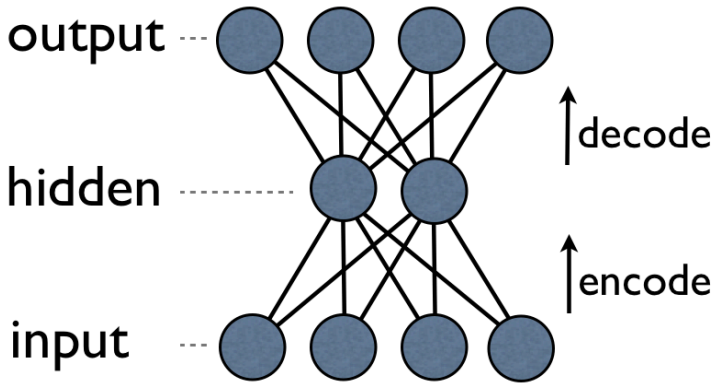
Autoencoder

- Feed-forward network trained to **replicate its input** (input is target signal for training) $\rightarrow$ **unsupervised method**



- Objective: minimize some form of reconstruction error
- Forces the network to learn a (compressed) **representation of the input** in hidden layers
  - For 1 hidden layer with k linear units, hidden neurons span the subspace of the first k principal components
  - With non-linear units, more complex representations can be learned
  - With stochastic units, corrupted input can be cleaned $\rightarrow$ **denoising autoencoder**

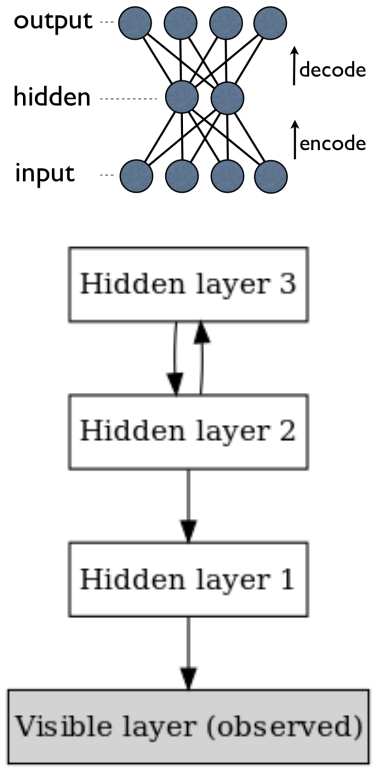
Autoencoder



- Dimensionality of the hidden layer can be **smaller** or **larger** than that of input/output
  - Smaller: yields a compressed representation (undercomplete)
  - Larger: results in a mapping to a higher-dimensional feature space (overcomplete)
- Typically trained with a form of stochastic gradient descent
- **Deep autoencoder:** # hidden layers > 1

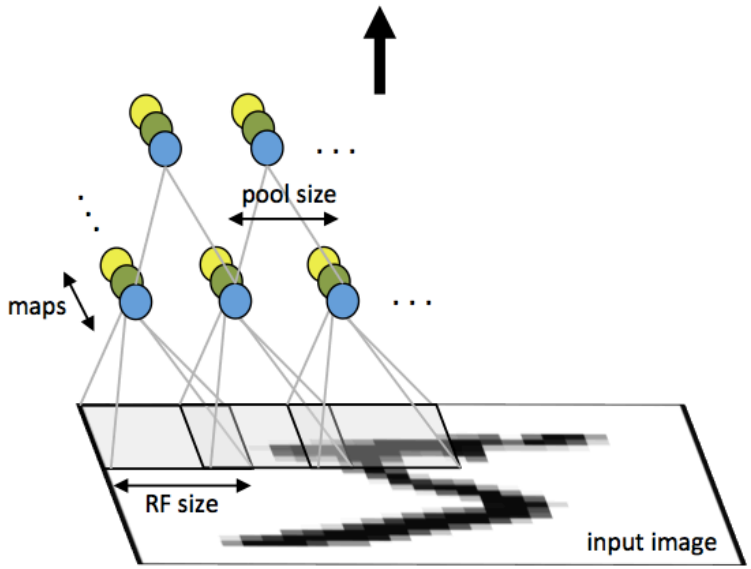
Stacked Autoencoder

- Autoencoder can be used as the learning module within a Deep Belief Network $\rightarrow$ **stacked autoencoders**
  1. Train the first layer as an autoencoder to minimize some form of reconstruction error of the raw input
  2. The hidden units' outputs (i.e., the codes) of the autoencoder are now used as input for another layer, also trained to be an autoencoder
  3. Repeat (2) until the desired number of additional layers is reached
- Can also be used for pre-training a network followed by fine-tuning via supervised learning



Convolutional neural network (CNN)

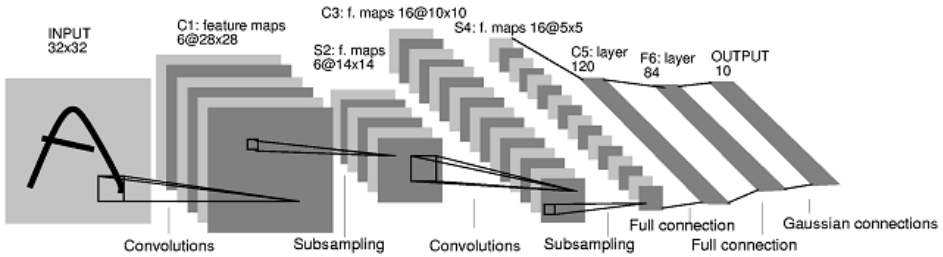
- Before DBNs, supervised deep neural networks have been difficult to train, with one exception: **convolutional neural networks** (CNNs)
- inspired by biological processes in the visual cortex



- **Topological structure:** neurons are arranged in filter maps that compute the same features for different parts of the input

Convolutional neural network (CNN)

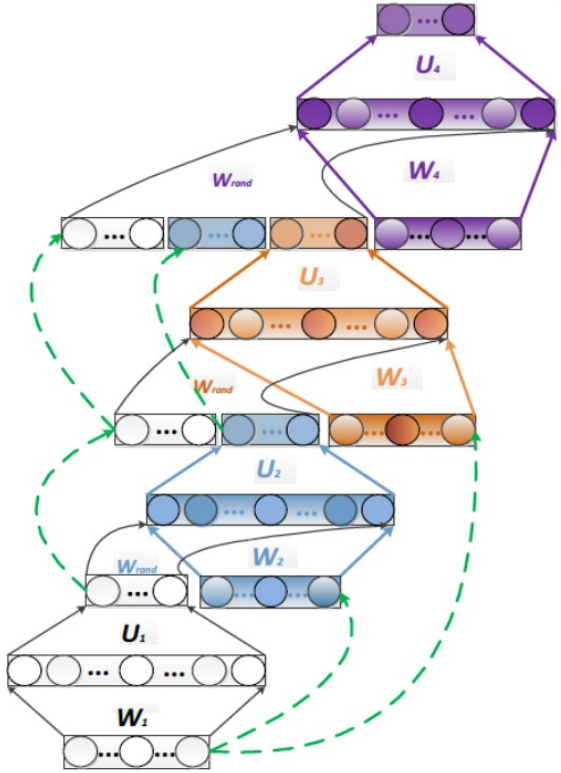
- Typical CNNs have 5-7 layers
- A CNN for handwritten digit recognition (LeCun et al., 1998):



- Reasons why standard gradient descent methods are tractable for CNNs:
  - **Sparse connectivity:** Neurons receive input only from a local receptive field (RF)
  - **Shared weights:** Each neuron computes the same function for each RF
  - **Pooling:** Predefined function instead of learnt weights for some layers, e.g. *max*

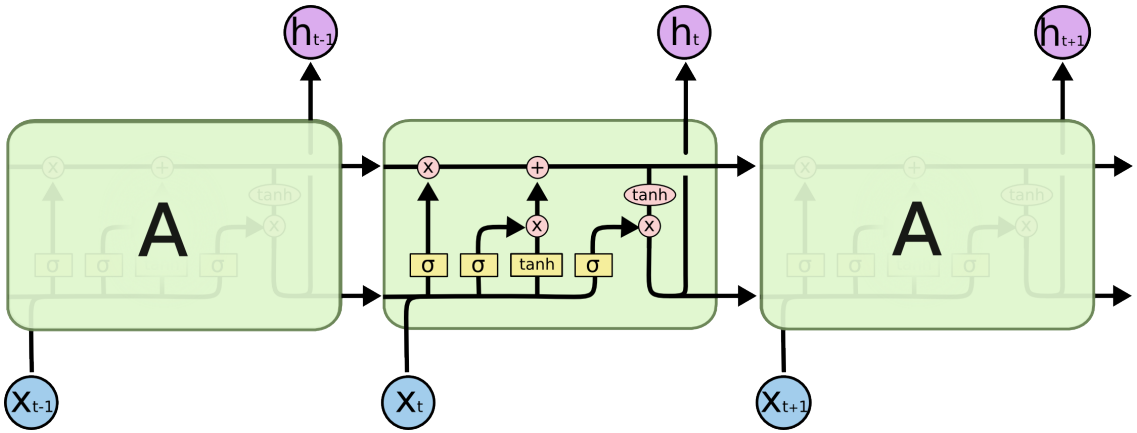
Deep stacking network (DSN)

- Simple classifiers are **stacked on top of each other** to learn a complex classifier
  - e.g., CRFs, two-layer networks
- Originally designed for scalability: simple classifiers can be **efficiently trained** (convex optimization $\rightarrow$ "deep convex network")
- Features for a classifier at a higher level are a **concatenation** of the classifier outputs of lower modules and the raw input features



Long Short Term Memory Networks (LSTM)

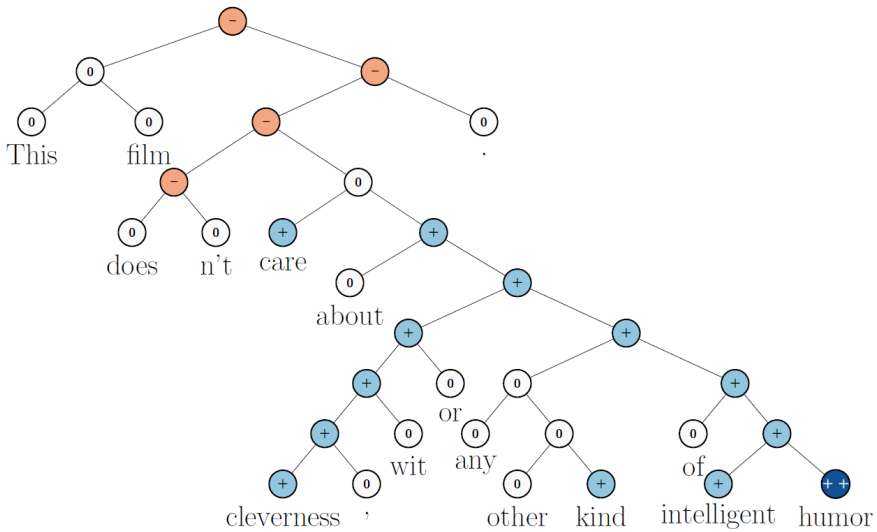
- **Recurrent neural networks** designed to learn long-term dependencies
- Keeps a state, which is controlled by a number of gates
- `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`

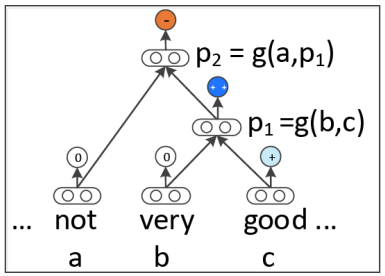


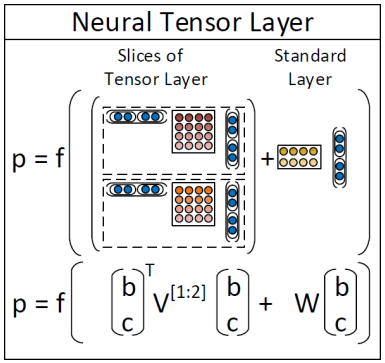
Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–80. http://doi.org/10.1162/neco.1997.9.8.1735

Recursive Neural Tensor Network (RNTN)

- **Tree structure** with a neural network at each node
- Used in **natural language processing**, e.g., for sentiment detection
- Socher et al., 2013: parse sentences into a binary tree, and at each node classify sentiment in a bottom-up manner (5 classes: – –, –, 0, +, ++)





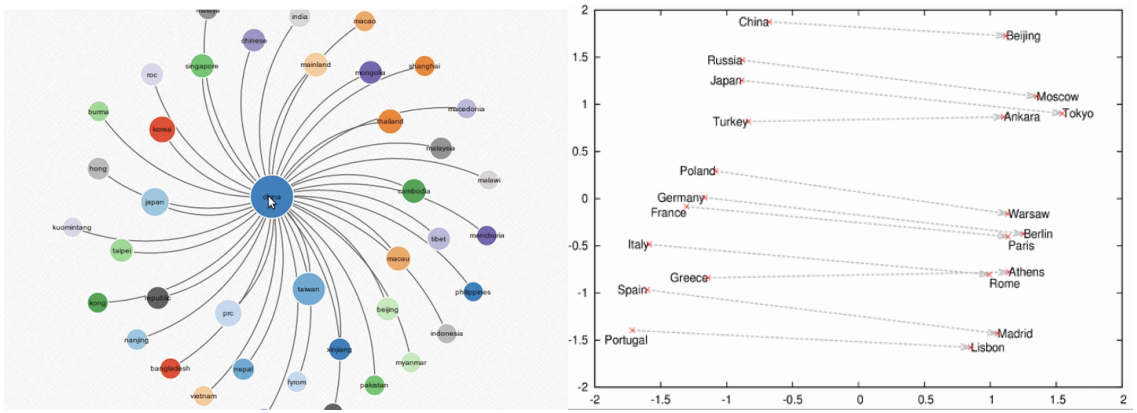


Deep learning with textual data

- Text has to be transformed into real-valued vectors that deep learning algorithms can understand
- **Word2Vec:** efficient algorithms developed by Google (`https://code.google.com/p/word2vec/`)
- Word2Vec itself is **not** deep learning (it uses shallow ML methods)
- Given a text it automatically learns relationships between words based on their context
- Each word is represented by a vector in a space where related words are close to each other, i.e. **word embedding**
- Word vectors can be used as features in many natural language processing and machine learning applications

Deep learning with textual data

- Interesting properties of Word2Vec vectors:
  - $vector('Paris') - vector('France') + vector('Italy') \approx vector('Rome')$
  - $vector('king') - vector('man') + vector('woman') \approx vector('queen')$



- Training is performed via a two-layer neural network (hierarchical softmax or negative sampling)
- Input (word context) is represented as continuous bag of words or skip-grams

Deep learning and reinforcement learning

- Reinforcement learning as an application for deep learning, e.g. AlphaGo
- Initially reinforcement learning worked only on discrete action spaces, e.g. DQN (Deep Q-Networks)
- Extension for continuous action spaces, e.g. DDPG (Deep Deterministic Policy Gradient)

Categorization of Deep Learning approaches

- **Deep networks for unsupervised learning**
  - e.g., Restricted Boltzmann Machines, Deep Belief Networks, autoencoders, Deep Boltzmann machines, ...
- **Deep networks for supervised learning**
  - e.g., Convolutional Neural Networks, Deep Stacking Networks, ...
- **Hybrid deep networks:** make use of both unsupervised and supervised learning (e.g., "pre-training")
  - e.g., pre-training a Deep Belief Network composed of Restricted Boltzmann Machines

Alternative Deep Learning Architectures

Deep learning is not limited to neural networks

- Stacked SVMs with random projections
  Vinyals, Ji, Deng, & Darrell. Learning with Recursive Perceptual Representations.
  `http://books.nips.cc/papers/files/nips25/NIPS2012_1290.pdf`
- Sum-product networks
  Gens & Domingos. Discriminative Learning of Sum-Product Networks.
  `http://books.nips.cc/papers/files/nips25/NIPS2012_1484.pdf`

How to choose the right network?

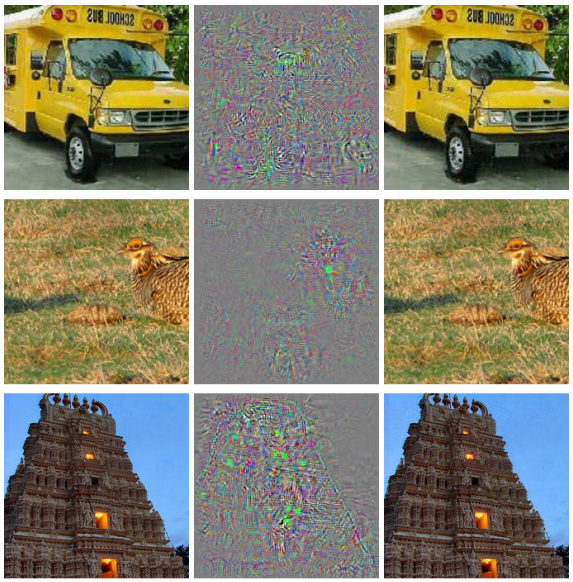
Data Sector	Use Case	Input	Transform	Neural Net
Text	Sentiment analysis	Word vector	Gaussian Rectified	RNTN or DBN (with moving window)
	Named-entity recognition	Word vector	Gaussian Rectified	RNTN or DBN (with moving window)
	Part-of-speech tagging	Word vector	Gaussian Rectified	RNTN or DBN (with moving window)
	Semantic-role labeling	Word vector	Gaussian Rectified	RNTN or DBN (with moving window)
Document	Topic modeling/ semantic hashing (unsupervised)	Word count probability	Can be Binary	Deep Autoencoder (wrapping a DBN or SDA)
	Document classification (supervised)	TF-IDF (or word count prob.)	Binary	Deep-belief network, Stacked Denoising Autoencoder
Image	Image recognition	Binary	Binary (visible and hidden)	Deep-belief network
		Continuous	Gaussian Rectified	Deep-belief network
	Multi-object recognition			Convolutional Net, RNTN (image vectorization forthcoming)
	Image search/ semantic hashing		Gaussian Rectified	Deep Autoencoder (wrapping a DBN)
Sound	Voice recognition		Gaussian Rectified	Recurrent Net
				Moving window for DBN or ConvNet
Time Series	Predictive analytics		Gaussian Rectified	Recurrent Net
				Moving window for DBN or ConvNet

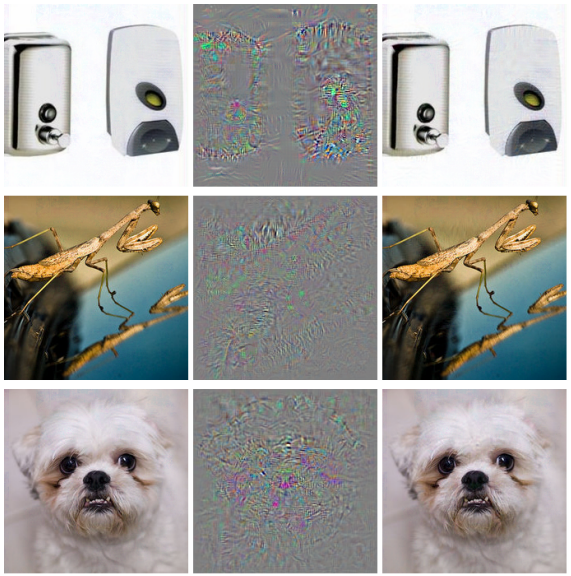
`http://deeplearning4j.org/neuralnetworktable.html`

Limitations of Deep Learning

Limitations

- Team at Google made an interesting finding
- Small changes in the input yield an big, “unexpected” change in the output
- Left images are labelled correctly, the right images are misclassified, the image in the centre the shows the difference between the images





Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

Available Software Toolkits

Available toolkit to get started

- TensorFlow
  - ... often used with Keras
- Theano
- Torch
- Deeplearning4J (Java)
- 0xdata / H2O

Resources

- Y. Bengio (2009). Learning Deep Architectures for AI. Foundations and Trends in Machine Learning, 2(1), 1–127.
- L. Deng and D. Yu (2014). Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing, 7(3-4), 197–387.
- J. Schmidhuber (2014). Deep Learning in Neural Networks: An Overview. http://arxiv.org/abs/1404.7828.
- https://cs.stanford.edu/people/karpathy/convnetjs/
- http://deeplearning.net/
- http://en.wikipedia.org/wiki/Deep_learning
- etc.

The End

Next: Q&A Session III (HS i5)

Acknowledgements: Many thanks to Stefan Klampfl for preparing the slides!