Computational aspects of machine learning

www.huawei.com

Gunnar Peters
Huawei Sweden

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Content

- Huawei Sweden Algorithm Group
- Introduction to learning
- Training neural networks
- Machine Learning in Radio Resource Management

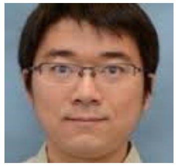
Huawei Sweden Algorithm Group

HUAWEI TECHNOLOGIES CO., LTD.

The Algorithm Group Huawei Sweden

Gunnar Peters
Technical lead
Algorithm Group
Huawei Sweden

5G RRM
RRM architecture
Traffic steering
Machine Learning
(Pablo Soldati, KTH)

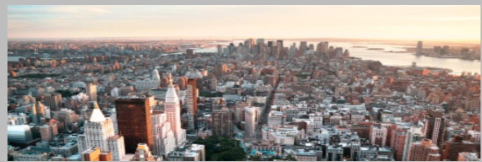


4G RRM
Coordinated scheduling
Small packet optimization
(Xiaojia Lu, CWC Oulo)

Baseband Receiver
Receiver architecture
Mac / Phy codesign
(Jinliang Huang, KTH)

Resource allocation in Wireless networks

Complexity of the system

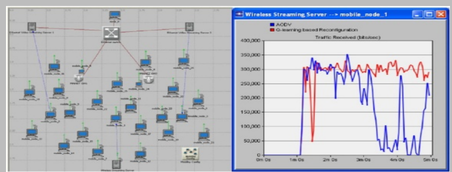


$$\cancel{\log 2(\det(Id + H^{H} \cdot H))}$$

Quantized modulation and coding

- Simplified models
- Imperfect knowledge

We resort to simulations



- Use of simplified control rules with tunable parameters
- Algorithms and parameter values tuned by simulations

- We do not have exact information of radio conditions
- Traffic load always changing
- No straight forward closed form rule mapping e.g. radio conditions to radio performance

RRM is an optimal control problem where the underlying dynamics are not known

No precise dynamic model

Machine Learning as a solution

No precise dynamic model

Too many variables

- The obvious state space for a Radio Access Network consists of, for example, radio channels, traffic and KPIs for all users in the network.
- We are back in the curse of dimensionality

•Big data

- There are efficient methods for automatic extraction of essential features.
- Extremely successful in computer vision and language processing.
- Statistical learning, deep neural networks etc

•Reinforcement Learning

- Adaptive optimal control
- Model free
- Learns the value of different actions through interaction with the system
- Used successfully in robotics, self driving cars and other Artificial Intelligence applications

Yes

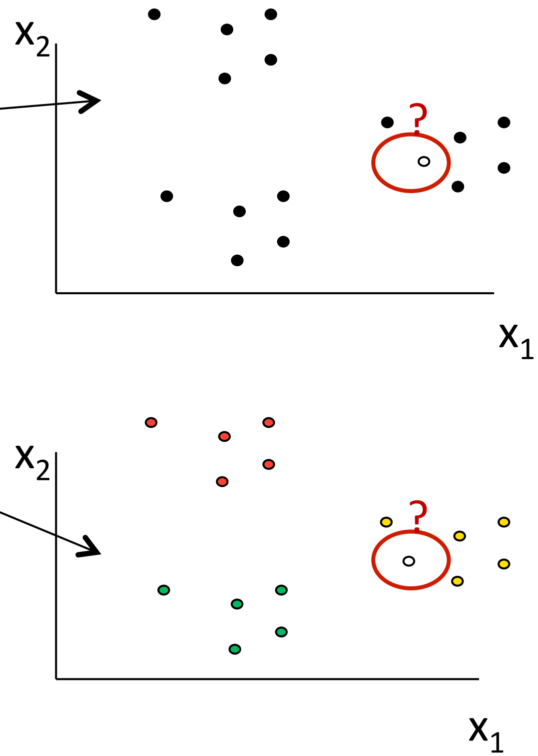
Introdcution to learning

HUAWEI TECHNOLOGIES CO., LTD.

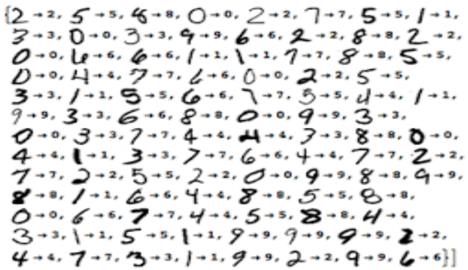
HUAWEI

Classification of machine learning problems

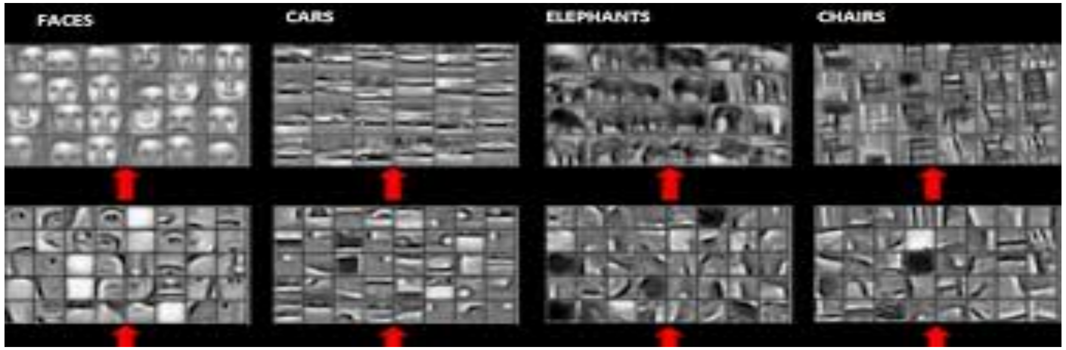
- **Unsupervised learning**
  - Find structure in data
  - Reduce dimension of data
- **Supervised learning**
  - Labeled data
  - Predict the label of new data
- **Reinforcement learning**
  - Learning optimal control by interacting with a system



Example of supervised learning



Recognizing hand written digits

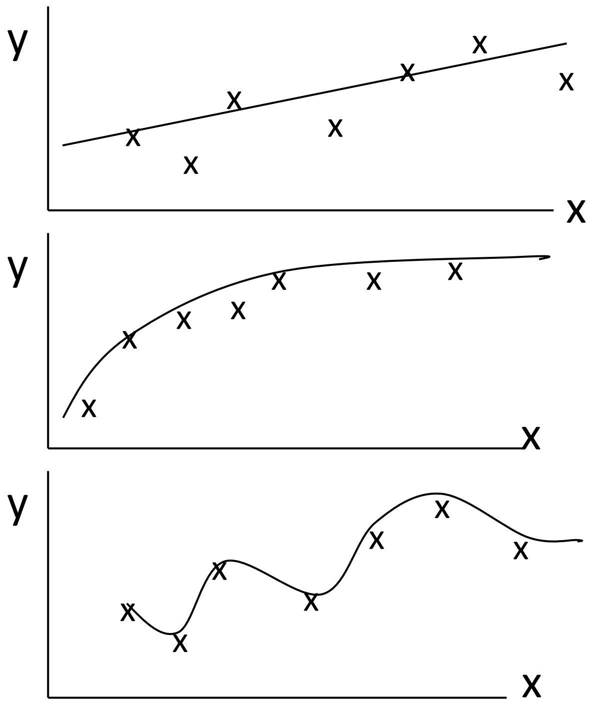


Class Name	# Images
Bear	105
Cougar	100
Cow	97
Coyote	100
Deer	100
Elephant	100
Giraffe	84
Horse	100
Kangaroo	90
Leopard	100
Lion	98
Panda	97
Penguin	80
Sheep	68
Skunk	62
Tiger	100
Zebra	77

Classifying animals

Deep learning uses neural networks to learn hierarchical features on general images. These features simplify the classification.

Learning = regression



- Fitting a model in noise
- Have to learn the model order
  - Low model order -> model error
  - High model order -> over fitting
- We need to learn models with good generalization properties

Higher dimensions

This fairly easy in low dimension

- There are good methods for learning model orders

In higher dimensions (n > 500)

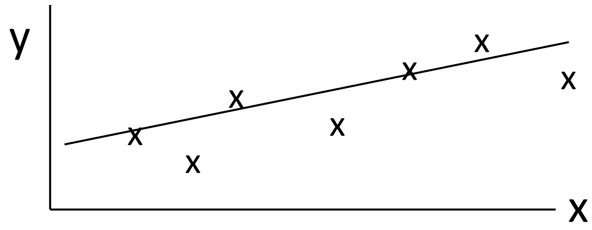
- Complex problem
- Number of learning samples needed grow exponentially with the dimension

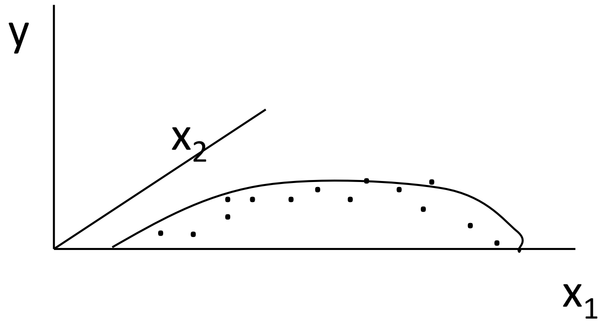
In most problem there is a hidden structure in the data

More advanced learning methods

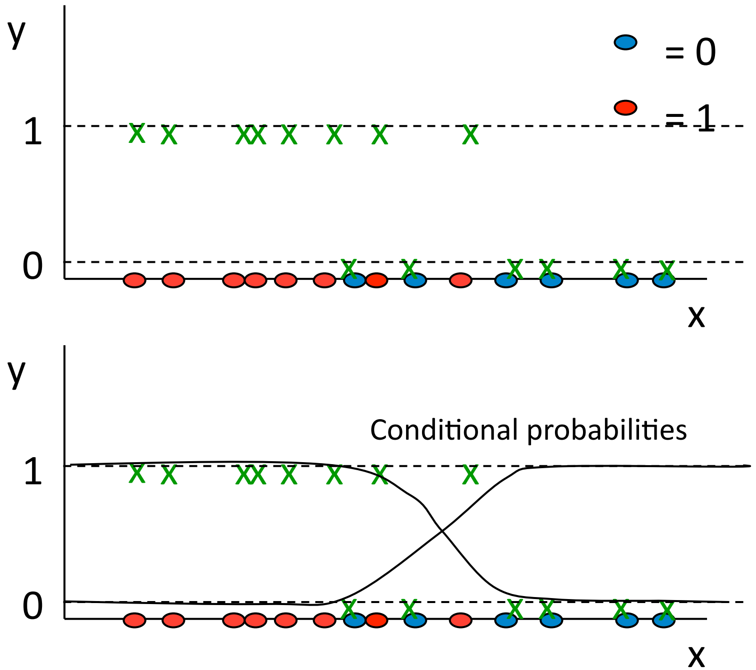
- Neural networks
- Decision trees
- Gaussian processes

Identifies hidden structures and restricts the learning to a low dimensional manifold (hyper surface)





Formulation of classification problems



- Classification is formulated as a regression of likelihood functions.
- Instead of fitting the data to hard values (e.g. 0 and 1) we use conditional probabilities, which are supposed to vary smoothly with x.

$$P(\text{blue} \mid x)$$

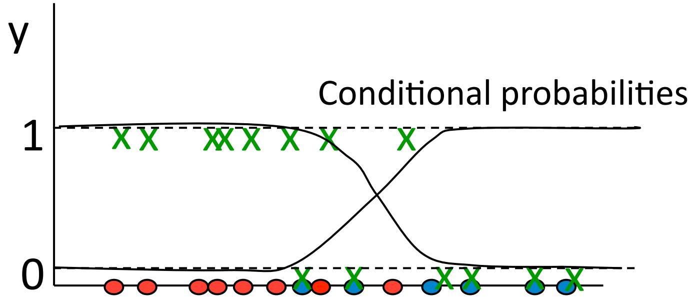
$$P(\text{red} \mid x)$$

Training neural networks

HUAWEI TECHNOLOGIES CO., LTD.

HUAWEI

Models



Linear model

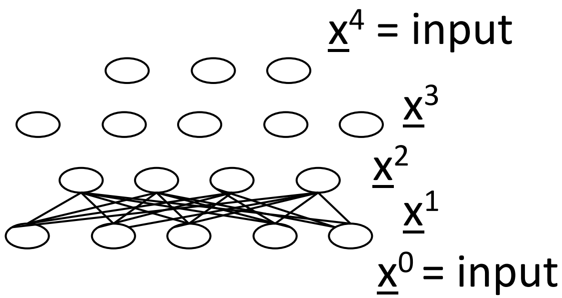
$$y = \alpha \cdot x + \beta$$

Sigmoid

$$y = \sigma(\alpha \cdot x + \beta)$$

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Neural network



Generalization to higher dimensions and model orders

$$\underline{x}^{i+1} = \sigma(\underline{W}^{i} \cdot \underline{x}^{i} + \underline{m}^{i})$$

output vector at layer *i+1*

Weight matrix

bias

Learning as optimization

output input

$$(y^{(j)}, x^{(j)}), \quad j = 1, \ldots, M$$

Learning samples obtained by interaction with the real world. $x^{(j)}$ and $y^{(j)}$ are vectors.

$$f(W,m) = \frac{1}{M} \sum_j \left\| y^{(j)} - nn(x^{(j)}, W, m) \right\|^2$$

Loss function. ***nn*** is the approximation defined by the neural network.

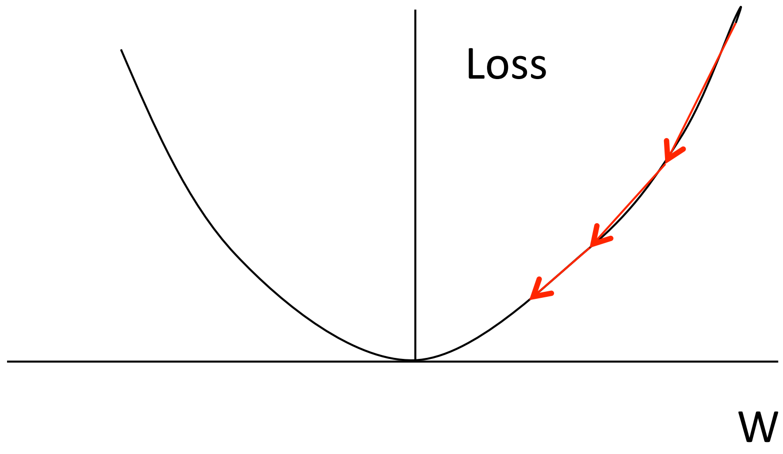
Learning now becomes the optimization problem

$$\min_{W,m} f(W,m)$$

Gradient descent

$$m(n+1) = m(n) - \alpha \cdot grad_m(f)$$

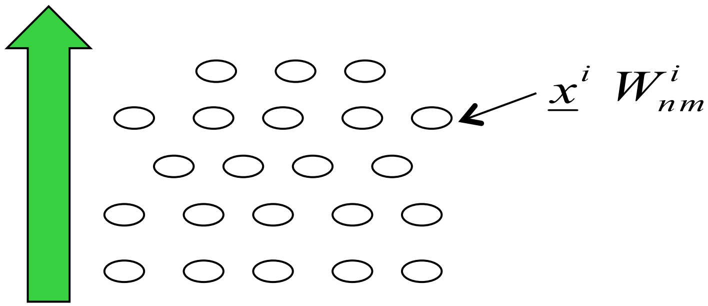
$$W(n+1) = W(n) - \alpha \cdot grad_W(f)$$



Backprop algorithm for calculating the gradient

Forward propagation of values

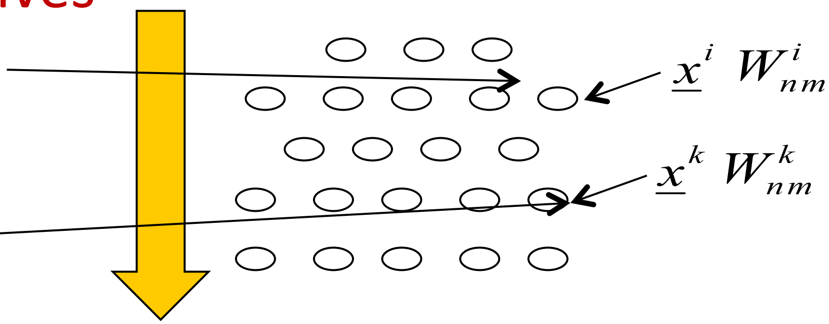
$$\underline{x}^{i+1} = \sigma(W^i \cdot \underline{x}^i + m^i)$$



Backward propagation of derivatives

$$\frac{\partial \underline{x}_a^{i+1}}{\partial W^k_{nm}} = \sigma'(W^i \cdot \underline{x}^i + m^i) \cdot W^i_{ab} \cdot \frac{\partial \underline{x}_b^i}{\partial W^k_{nm}}$$

$$\frac{\partial \underline{x}_n^{k+1}}{\partial W^k_{nm}} = \sigma'(W^k \cdot \underline{x}^k + m^k) \cdot \underline{x}_n^i$$



Putting it all together

The gradient of the loss function

$$f(W,m) = \frac{1}{M}\sum_{k}\left\| y^{k} - nn(x^{k},W,m) \right\|^{2}$$

Can now be calculated using the backprop algorithm

$$\frac{\partial f(W,m)}{\partial W_{nm}^{k}} = -\frac{1}{M}\sum_{j \in Training\,samples}(y^{(j)} - nn(x^{(j)},W,m)) \cdot \frac{\partial nn(W,m)}{\partial W_{nm}^{k}}$$

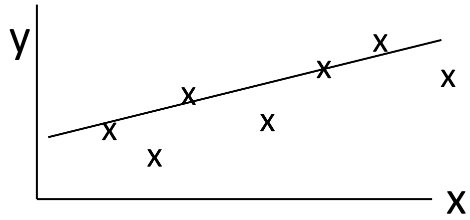
Using e.g. the gradient descent method

$$W(n+1) = W(n) - \alpha \cdot grad_{W}(f)$$

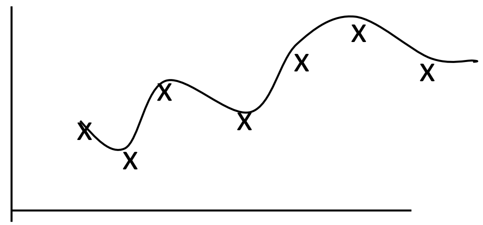
We have a computational scheme for learning from training samples

Generalization

- We are only training on a subset of, for example, images of animals.
- There is a chance that what we learn only applies to this subset and does not generalize.
- It is not always best to iterate until the loss function is as small as possible.



Training error large
Generalization good



Training error small
Generalization poor

Stochastic gradient descent

Writing the loss function as a sum over the training samples

$$f(W,m) = \frac{1}{M}\sum_j \left\| y^{(j)} - nn(x^{(j)},W,m) \right\|^2 = \frac{1}{M}\sum_j f^{(j)}(W,m)$$

The gradient can be written

$$grad\left(\frac{1}{M}\sum_j f^{(j)}(W,m)\right)$$

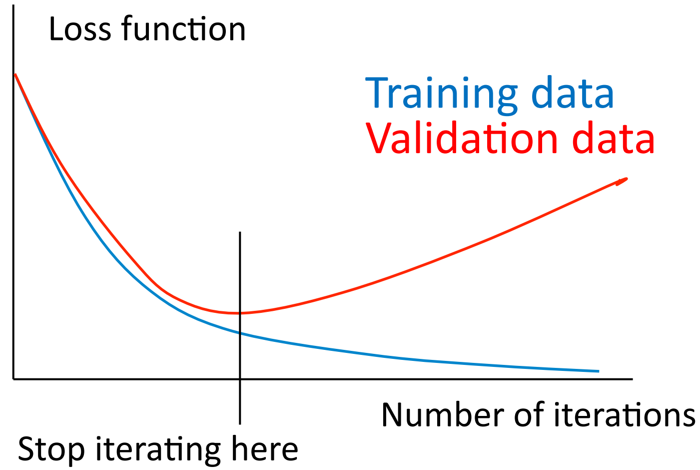
The stochastic gradient descent means randomly picking training samples and update the parameters according to

$$W = W - \alpha \cdot grad(f^{(j)}(W,m))$$

Theory of random iterations now guarantees good generalization

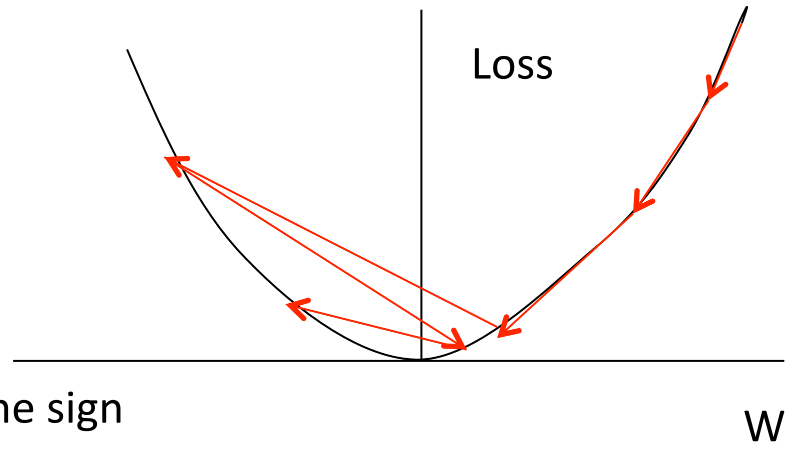
Convolutional neural networks

- **Still very sensitive to thing like step size and topology of the neural network.**
- **Test generalization on a subset of available data**



Rprop: Special iteration methods

- **Rprop:**
  - Adaptive step size
  - Different step size for each parameter
  - Gradient gives direction but not size of the incremental improvements

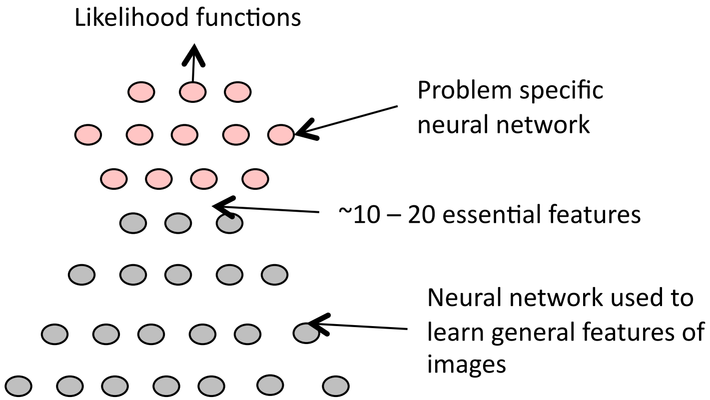


Increase ΔW while partial derivative has same sign
Decrease when it changes sign

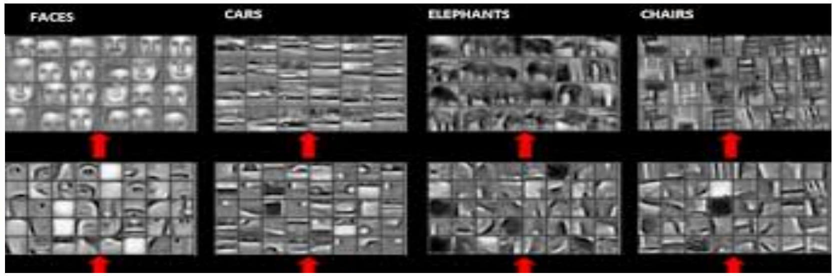
Very stable
Does not avoid generalization problem

Deep neural networks

- Image data is typically concentrated around low dimensional manifold (not every 128 by 128 array is an image).
- In deep learning the neural network is split into a lower part with only a few outputs, and a top part which is problem specific.
- The lower part is generic and reused between different sets of images and different problems. It can therefore be trained on an ever increasing set of training samples.
- The top part is problem specific, but since the input dimension is low smaller sets of data can be used..



Class Name	# Images
Bear	105
Cougar	100
Cow	97
Coyote	100
Deer	100
Elephant	100
Giraffe	84
Horse	100
Kangaroo	90
Leopard	100
Lion	98
Panda	97
Penguin	80
Sheep	68
Skunk	62
Tiger	100
Zebra	77



Other types of neural networks

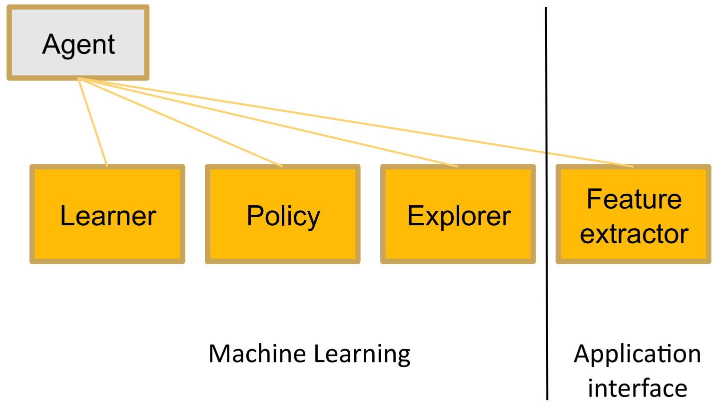
- **Deep Neural Networks (DNN)**
  - Tries to auto encode the data into a few features
- **Convolutional Neural Networks (CNN)**
  - Uses translation invariance of images to reduce the number of weights in the network.
  - Same weights reused on different parts of the image
- **Recurrent Neural Networks (RNN)**
  - Feed back loops between the layers are used to introduce memory in the network

Machine Learning in Radio Resource Management

HUAWEI TECHNOLOGIES CO., LTD.

Huawei Sweden machine Learning software

- The machine learning simulator framework consists of the following components
    - Machine Learning
        - Learner
        - Policy
        - Agent
        - Explorer
    - Interface to application
        - Feature extractor



- In e.g. the Learner component there will be there will be classes for
    - Neural Network,
    - Decision Tree
    - Ensemble learners

respectively.

- The Machine Learning part is application independent. The application specific code ´will reside in the implementation classes of the Feature Extractor component. In this way the Machine Learning software can be used in any simulator or program with minimal effort.

Highlights of machine learning in RRM

- **This framework has been applied to different 3G and 4G use cases**
  - 3G RoT adaptation
    - 25 % capacity gain with no loss in cell edge performance
  - HetNet Cell Range Expansion
    - Up to 80% gain in hotspot scenarios (where CRE is supposed to improve the performance) with little loss in other scenarios.
  - 4G Uplink power control
    - Around 100% gain. This depends on the scenario.
  - 4G TX power allocation
    - 200 % reduction in TX power
  - 4G Single Frequency Network threshold tuning
    - 100 – 150 % gain in high load

References

1. http://deeplearning.net/tutorial/
2. Deep Learning, Yoshua Bengio, Ian Goodfellow, Aaron Courville, MIT Press.
3. Yoshua Bengio, Learning Deep Architectures for AI, Foundations and Trends in Machine Learning, 2(1), pp.1-127, 2009.
4. Yoshua Bengio, Aaron Courville, Pascal Vincent, Representation Learning: A Review and New Perspectives, Arxiv, 2012.
5. Jurgen Schmidhuber, Deep Learning and Neural Networks: An Overview, arXiv, 2014.
6. Leon Bottou, Frank E. Curtisy, Jorge Nocedalz, Optimization Methods for Large-Scale Machine Learning, arXiv: 1606.04838v1 [stat.ML] 15 Jun 2016.
7. Moritz Hardt, Benjamin Recht, Yoram Singer , Train faster, generalize better, stability of stochastic gradient descent, arXiv.org > cs > arXiv:1509.01240
8. Francesco Davide Calabrese; Euhanna Ghadimi; Leo Wang; Gunnar Peters, Pablo Soldati, Learning Radio Resource Management in 5G Networks: Framework, Opportunities and Challenges, submitted IEEE Communications Magazine

Thank you

www.huawei.com