Machine Learning Introduction and Exemplary Application in Embedded Wireless Platforms

Jonathan Ah Sue – Intel Germany GmbH

Agenda

- Machine learning (ML) fundamentals
  - A brief history of artifical intelligence
  - The five tribes of machine learning
  - The tasks of ML algorithms
  - Regression as illustrative example
  - Deeper on neural networks
  - Generalized linear models
  - Theoritical machine learning
- Cognitive power control: ML in practice

A brief history of artifical intelligence

- Inspired from 3 different fields (McCulloh and Pitts, 1943) [1]:
  - Functions of biological neurons
  - Formal analysis (Russel and Whitehead)
  - Theory of computation (Alan Turing)
- Computing Machinery and Intelligence (Alan Turing, 1950) [2]:
  - Turing test, machine learning, genetic algorithms and reinforcement learning
- Darmouth seminar (John McCarthy, 1956) [3]:
  - Artifical intelligence, Logic Theorist (LT) of Newell and Simon
- Perceptron (Frank Rosenblatt, 1962) [4]:
  - Convergence theorem (Block et al., 1962) [5]

A brief history of artifical intelligence

Machine learning fundamentals

- The AI failures or „AI winter“:
  - Machine translation (1966) [6], automatic theorem proof, Lighthill report (1973) [7], limited representation capabilities of perceptrons
- Expert systems (1980 – early 1990s):
  - DENDRAL program (Buchanan et al., 1969) [8], MYCIN program for blood diseases comparable to domain experts (450 simple rules) [9], first commercial success with XCON (1980)
  - Expensive and difficult to maintain
- Backpropagation [10]:
  - Steepest descent with chain rule (Bryson et al., 1969)
  - First neural network application (Werbos, 1982)

accellera SYSTEMS INITIATIVE

A brief history of artifical intelligence

- AI as a science (1990s):
  - Methodology driven by rigourous statistical analysis (Cohen, 1995) [11]
  - Hidden Markov models (speech recognition), information theory (automatic translation), Bayesian networks (reasoning), support vector machines, random forest...
- Recent achievements:
  - Audio: speech recognition based on LSTM (Hochreiter, Schmidhuber and Gers), lip reading, audio generation
  - Image/video: OCR with CNN, *cat network* recognition in videos (2012) [12], self-driving cars
  - Generative adversarial networks, reinforcement learning
- Communications, data availability, computational power, better algos
- Does the AI world run on neural networks?

The five tribes of machine learning

Machine learning fundamentals

- Inspiration and source of knowledge:
  - Evolution, experience, culture, computers
- Paradigms of ML (Pedro Domingos, 2015) [13]:

Tribe	Origins	Master Algorithm
Symbolists	Logic, philosophy	Inverse deduction
Connectionnists	Neuroscience	Backpropagation
Evolutionnaries	Evolutionary biology	Genetic programming
Bayesians	Statistics	Probabilistic inference
Analogizers	Psychology	Kernel machines

The tasks of ML algorithms

Machine learning fundamentals

- Learning tasks:
  - Supervised: What is the best mapping function between inputs and outputs?
  - Unsupervised: What makes 2 samples similar?
  - Semi-supervised: Can we cluster unlabelled data and learn efficiently under this uncertainty?
  - Reinforcement learning: Given the rules and the goal to achieve, how can I optimize myself?
- Numerical data type:
  - Classification / regression
- Statistical data type:
  - Binary / categorical / ordinal / binomial / count / real-valued additive / real-valued multiplicative
- Multivariate and/or multidimensional

Regression as illustrative example

Machine learning fundamentals

- Roadmap:
  - Least square regression
  - Gradient descent
  - Maximum likelihood
  - Maximum a posteriori
  - Bayesian linear regression
  - Gaussian process

Least square regression

Machine learning fundamentals > Regression as illustrative example

- Training set: $D = \{(\mathbf{x}^{(j)}, y^{(j)})\}_{j=1}^{m}$

  Input Target

- Hypothesis: $\mathbf{x} \rightarrow h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T\mathbf{x} = \sum_{i=1}^{n} \theta_i x_i$

- Cost function: $e(\boldsymbol{\theta}) = \frac{1}{2}\sum_{j=1}^{m}\left[h_{\boldsymbol{\theta}}(\mathbf{x}^{(j)}) - y^{(j)}\right]^2$

  Output

- Objective: $\boldsymbol{\theta}_{min} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}}\, e(\boldsymbol{\theta})$

- Normal equations:

Given

$$\mathbf{X} = \begin{pmatrix} x_1^{(1)} & \cdots & x_n^{(1)} \\ \vdots & \ddots & \vdots \\ x_1^{(m)} & \cdots & x_n^{(m)} \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{pmatrix}$$

, the analytical solution is

$$\boxed{\boldsymbol{\theta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

Gradient descent

Machine learning fundamentals > Regression as illustrative example

- Update rule for one training sample:

$$\theta_i \coloneqq \theta_i - \alpha \frac{\partial}{\partial \theta_i} e(\boldsymbol{\theta}) \qquad \frac{\partial}{\partial \theta_i} e(\boldsymbol{\theta}) = \frac{\partial}{\partial \theta_i} \frac{1}{2} [h_{\boldsymbol{\theta}}(\mathbf{x}) - \mathrm{y}]^2 = [h_{\boldsymbol{\theta}}(\mathbf{x}) - \mathrm{y}] x_i$$

Learning rate

$$\theta_i \coloneqq \theta_i - \alpha \left[h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(j)}\right) - \mathrm{y}^{(j)}\right] x_i^{(j)} \quad , \forall\, i \in [\![1, n]\!]$$

- Multiple training samples:
  - Batch:

$$\theta_i \coloneqq \theta_i - \alpha \sum_{j=1}^{m} \left[h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(j)}\right) - \mathrm{y}^{(j)}\right] x_i^{(j)}$$

  - Stochastic (incremental):

For $j := 1$ to m

$$\theta_i \coloneqq \theta_i - \alpha \left[h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(j)}\right) - \mathrm{y}^{(j)}\right] x_i^{(j)}$$

Maximum likelihood

Machine learning fundamentals > Regression as illustrative example

- Probabilistic interpretation: $y^{(j)} = \boldsymbol{\theta}^T \mathbf{x}^{(j)} + \varepsilon^{(j)}$ $\quad \varepsilon^{(j)} \sim N(0, \sigma^2)$ IID

$$p(y^{(j)}|\mathbf{x}^{(j)}, \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left(y^{(j)} - \boldsymbol{\theta}^T \mathbf{x}^{(j)}\right)^2}{2\sigma^2}\right)$$

- Maximum likelihood: $\boldsymbol{\theta}_{MLE} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\ p(D|\boldsymbol{\theta})$

Maximize $\quad L(\boldsymbol{\theta}) = \prod_{j=1}^{m} p\left(y^{(j)}|\mathbf{x}^{(j)}, \boldsymbol{\theta}\right)$

Maximize $\quad l(\boldsymbol{\theta}) = \log L(\boldsymbol{\theta}) = m \log \frac{1}{\sqrt{2\pi}\sigma} - \underbrace{\frac{1}{2\sigma^2} \sum_{j=1}^{m} \left(y^{(j)} - \boldsymbol{\theta}^T \mathbf{x}^{(j)}\right)^2}_{\text{To minimize}}$

To minimize
(Least square equivalent to MLE + Gaussian noise model)

Maximum a posteriori

Machine learning fundamentals > Regression as illustrative example

- MAP estimator:

$$\boldsymbol{\theta}_{MAP} = \underset{\boldsymbol{\theta}}{\text{argmax}}\ \underbrace{p(\boldsymbol{\theta}|D)}_{\text{Parameter posterior}} = \underset{\boldsymbol{\theta}}{\text{argmax}}\ \frac{p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)}$$

Likelihood
Prior
Marginal likelihood
Parameter posterior

- Univariate case:

Prior $$p(\theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(\theta-\mu)^2}{2\sigma^2}\right) \sim N(\mu, 1)$$

Maximize $$l(\boldsymbol{\theta}) = \log p(D|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})$$

$$= m \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \sum_{j=1}^{m} \underbrace{\left(y^{(j)} - \boldsymbol{\theta}^T \mathbf{x}^{(j)}\right)^2}_{\text{MLE}} + \log \frac{1}{\sqrt{2\pi}} - \frac{1}{2} \underbrace{(\theta-\mu)^2}_{\text{Prevent overfitting}}$$

- Regularization:
  - Ridge regression (L2), LASSO regression (L1), Elastic Net (L1+L2)

Bayesian linear regression

Machine learning fundamentals > Regression as illustrative example

- Goal:
  - For the moment, we only have a point estimate of $p(\boldsymbol{\theta}|D)$
  - We want to have an analytical form of $p(\boldsymbol{\theta}|D)$
- After some work (1-dim multivariate case):

Parameter posterior: $\boldsymbol{\theta}|D \sim N\left(\frac{1}{\sigma^2}\mathbf{A}^{-1}\mathbf{X}^T\mathbf{y}, \mathbf{A}^{-1}\right)$ with $\mathbf{A} = \frac{1}{\sigma^2}\mathbf{X}^T\mathbf{X} + \frac{1}{\tau^2}\mathbf{I}$ and $\boldsymbol{\theta} \sim N(\mathbf{0}, \tau^2)$

Posterior predictive (using $p(y_*|\mathbf{x}_*, D) = \int p(y_*|\mathbf{x}_*, \boldsymbol{\theta})\, p(\boldsymbol{\theta}|D)\, d\boldsymbol{\theta}$) :

$$\boxed{y_*|\mathbf{x}_*, D \sim N\left(\underbrace{\frac{1}{\sigma^2}\mathbf{x}_*^T\mathbf{A}^{-1}\mathbf{X}^T\mathbf{y}}_{\text{Normal equations when } \tau \to 0 \text{, everything is fine}},\ \mathbf{x}_*^T\mathbf{A}^{-1}\mathbf{x}_* + \sigma^2\right)}$$

Gaussian process

Machine learning fundamentals > Regression as illustrative example

- Goal:
  - For the moment, we have the posterior predictive distribution for a linear IO relationship
  - We want to be able to model any kind of IO relationship
- Definition:
  - A Gaussian Process (GP) is a collection of random variables. Any finite set of the collection follows a joint Gaussian distribution.
  - Notation: $f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ with k a covariance function (i.e., psd)
- Idea:
  - We compute a distribution over a function instead of a distribution over parameters
  - Direct link between the prior and the posterior predictive, no need to marginalize over parameters

(*) The demonstration requires some time

Gaussian process

Machine learning fundamentals > Regression as illustrative example

- Basic GP: $\begin{bmatrix}\mathbf{y}\\ \mathbf{f}^*\end{bmatrix} \sim N\left(0, \begin{bmatrix}\mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**}\end{bmatrix}\right)$ with $\mathbf{y} = \mathbf{f}$ the target vector and $\mathbf{f}^*$ the testing output (prediction)
- Noisy GP: $\begin{bmatrix}\mathbf{y}\\ \mathbf{f}^*\end{bmatrix} \sim N\left(0, \begin{bmatrix}\mathbf{K}+\sigma^2\mathbf{I} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**}\end{bmatrix}\right)$ with $\mathbf{y} = \mathbf{f} + \boldsymbol{\varepsilon}$ the target vector
- Using the multivariate Gaussian conditional distribution formula (*) :

$$\boxed{\mathbf{f}_*|\mathbf{x}_*, \mathbf{x}, \mathbf{y} \sim N(\mathbf{K}_*^T[\mathbf{K}+\sigma^2\mathbf{I}]^{-1}\mathbf{y}\,, \mathbf{K}_{**} - \mathbf{K}_*^T[\mathbf{K}+\sigma^2\mathbf{I}]^{-1}\mathbf{K}_*)}$$

- Covariance function (also called *kernels*):
  - Type: use the knowledge of inputs relationships (symmetry, ...)
  - Parameters: $\operatorname{argmax}\ p(\mathbf{y}|\mathbf{X})$ solved by gradient descent for example

Neural networks

Machine learning fundamentals

- Roadmap:
  - Generalized linear models
  - Logistic regression
  - Feed-forward neural networks
  - Bias-variance dilemma
  - Convolutional neural networks
  - Recurrent neural networks

Generalized linear models (1-dim)

Natural parameter
Sufficient statistic
Log partition function

- Exponential family
  - Class of distributions $p(y;\eta) = b(y)\exp\left(\eta^T T(y) - a(\eta)\right)$
  - Gaussian (1) $\eta = \mu$ (2) $T(y) = y$ (3) $a(\eta) = \eta^2/2$ (4) $b(\eta) = \left(1/\sqrt{2\pi}\right)\exp(-y^2/2)$
  - Bernoulli (1) $\eta = \log(\phi/(1-\phi))$ (2) $T(y) = y$ (3) $a(\eta) = \log(1+e^\eta)$ (4) $b(\eta) = 1$
- Generalized linear model assumptions
  - Exponential family: $y|\mathbf{x};\boldsymbol{\theta} \sim \text{ExponentialFamily}(\eta)$
  - Given x, we want to predict $E[T(y)|\mathbf{x};\boldsymbol{\theta}]$
  - Linear relationship (here 1-dim): $\eta = \boldsymbol{\theta}^T\mathbf{x}$

MLE computed by GD for 1 sample

$$\boxed{\frac{\partial l(\theta_i)}{\partial \theta_i} \propto -[E[T(y)|\mathbf{x};\boldsymbol{\theta}] - y]x_i}$$

- Hypothesis
  - Gaussian $h_{\boldsymbol{\theta}}(\mathbf{x}) = E[y|\mathbf{x};\boldsymbol{\theta}] = \mu = \eta = \boldsymbol{\theta}^T\mathbf{x}$
  - Bernoulli $h_{\boldsymbol{\theta}}(\mathbf{x}) = E[y|\mathbf{x};\boldsymbol{\theta}] = \phi = 1/(1+e^{-\eta}) = 1/\left(1+e^{-\boldsymbol{\theta}^T\mathbf{x}}\right)$

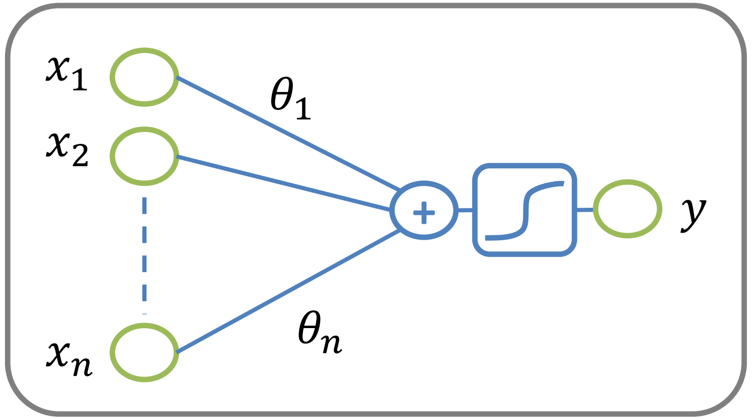
Logistic regression

- Bernoulli distribution $p(y|\mathbf{x};\boldsymbol{\theta}) = \phi^{\,y}(1-\phi)^{(1-y)}$
  - sigmoid as hypothesis $h_{\boldsymbol{\theta}}(\mathbf{x}) = 1/\left(1+e^{-\boldsymbol{\theta}^T\mathbf{x}}\right) = \phi$
  - logistic loss (cost) $e(\boldsymbol{\theta}) = \phi^{\,y}(1-\phi)^{(1-y)}$
- Same form for the GD (result as expected):

$$\theta_i := \theta_i - \alpha\left[h_{\boldsymbol{\theta}}\left(\mathbf{x}^{(j)}\right) - y^{(j)}\right]x_i^{(j)} \quad , \forall\, i \in [\![1, n]\!]$$

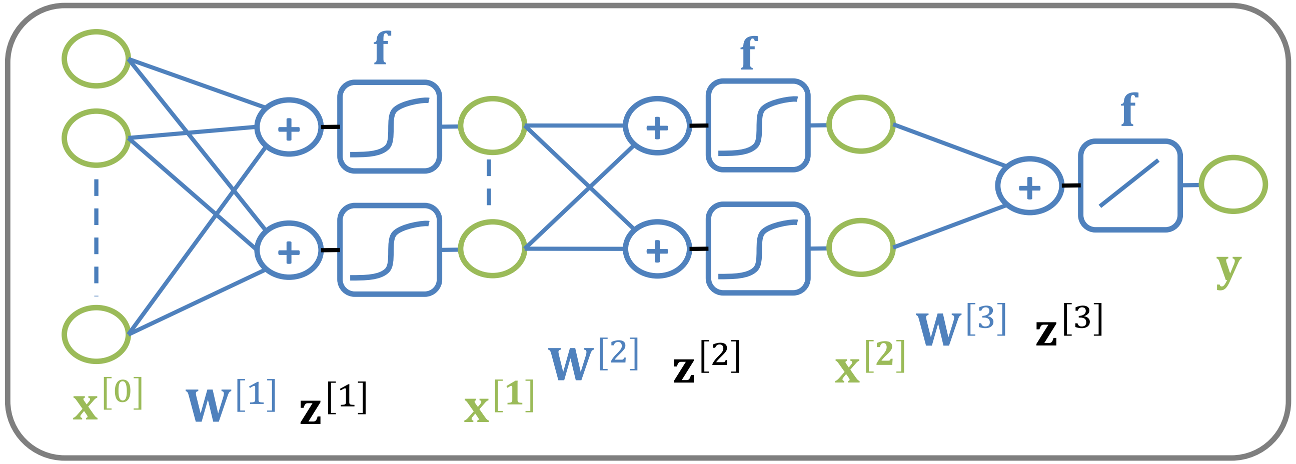
- Perceptron algorithm

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad \boldsymbol{\theta}^T\mathbf{x} \geq 0 \\ 0 & \text{if} \quad \boldsymbol{\theta}^T\mathbf{x} < 0 \end{cases}$$



- Newton (using the Hessian): $\boldsymbol{\theta} := \boldsymbol{\theta} - H^{-1}\,\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$

Feed-forward neural network



$$\mathbf{z}_n = \mathbf{W}_n \mathbf{x}_{n-1} \qquad \mathbf{x}_n = \mathbf{f}(\mathbf{z}_n)$$

- Backpropagation

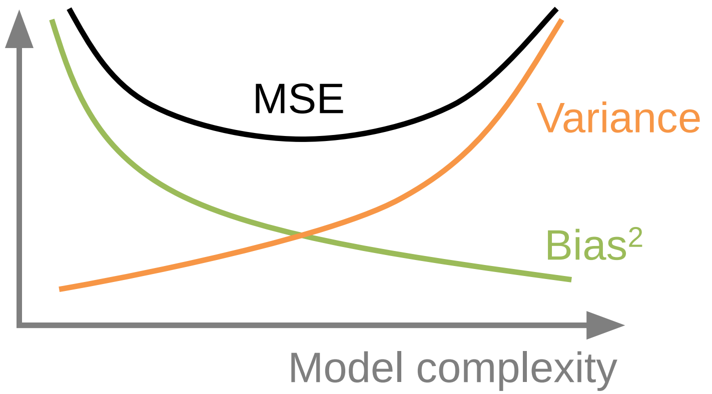
  Based on chain rule:

$$\frac{\partial e}{\partial \mathbf{W}^{[1]}} = \frac{\partial e}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{x}^{[2]}} \frac{\partial \mathbf{x}^{[2]}}{\partial \mathbf{x}^{[1]}} \frac{\partial \mathbf{x}^{[1]}}{\partial \mathbf{W}^{[1]}}$$

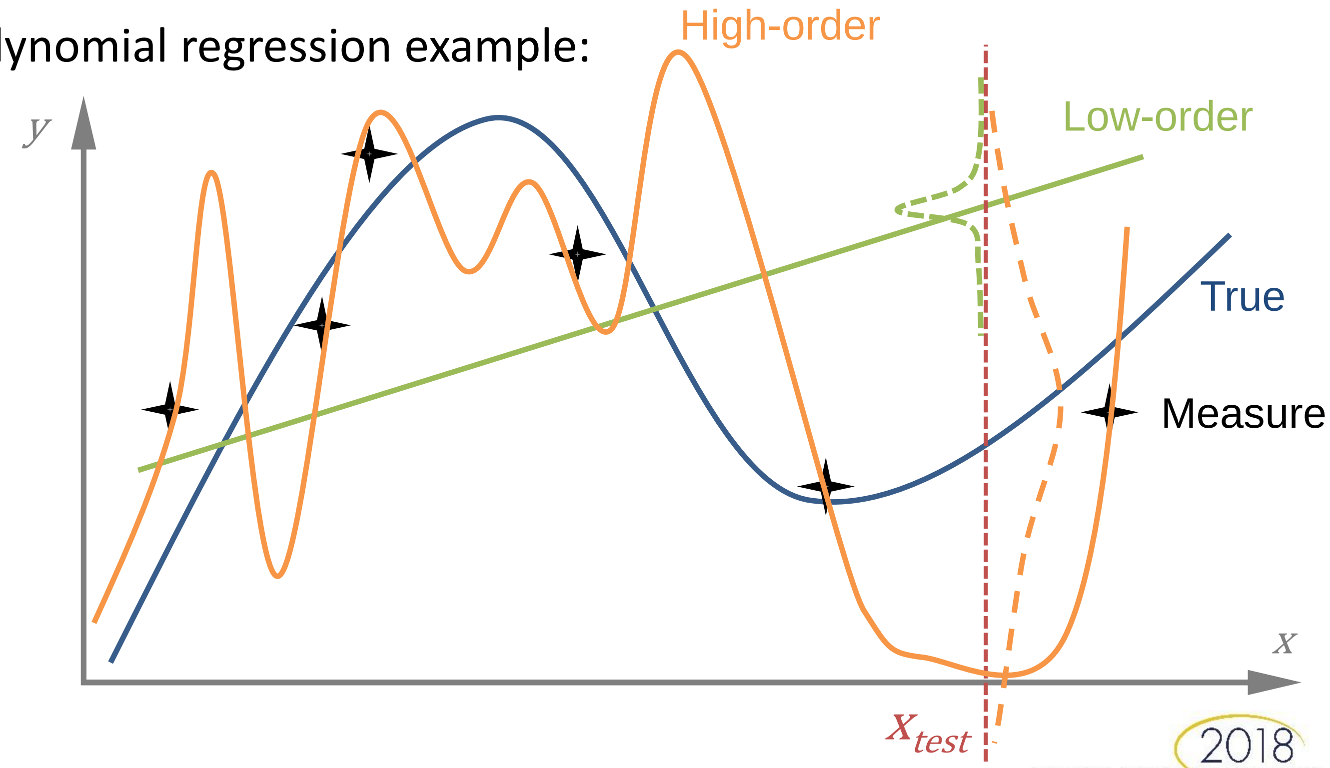
- Terms:
  - Weights, activation or transfer functions
- Universality:
  - Finite single hidden layer networks can theoritically compute any continuous function
- In practice:
  - Normalize and decorrelate inputs, tangent hyperbolic, learning rate per weight, momentum, seocnd-order methods, training and test set

Bias-variance dilemma

- Mean square error of an estimator $mse(\hat{y}) = E[(\hat{y} - y)^2 | y] = bias(\hat{y})^2 + var(\hat{y})$
- Solution (among others) for model selection



  – Polynomial regression example:

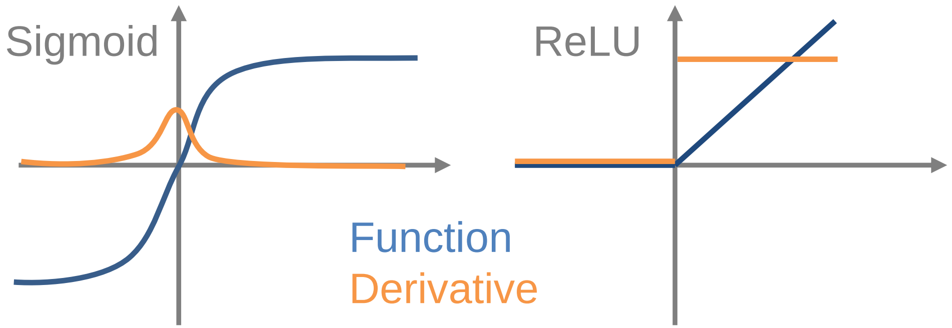


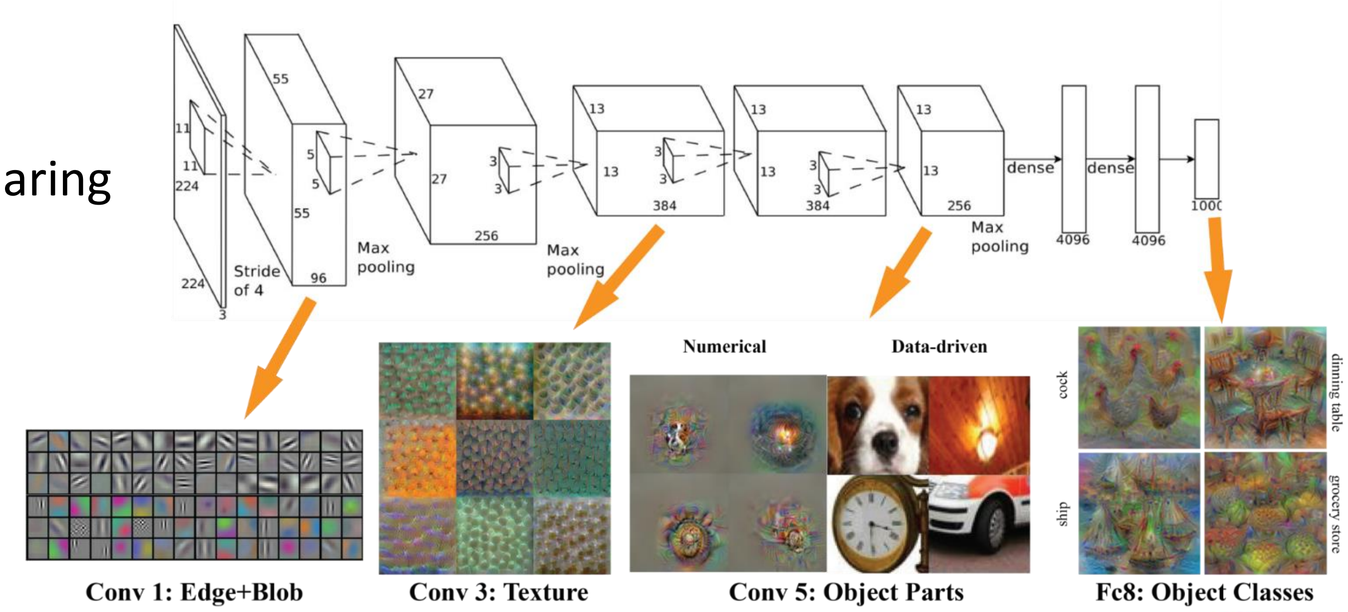
- For neural networks:
  – Training (70%) / validation (15%) / test (15%) split

Convolutional neural networks

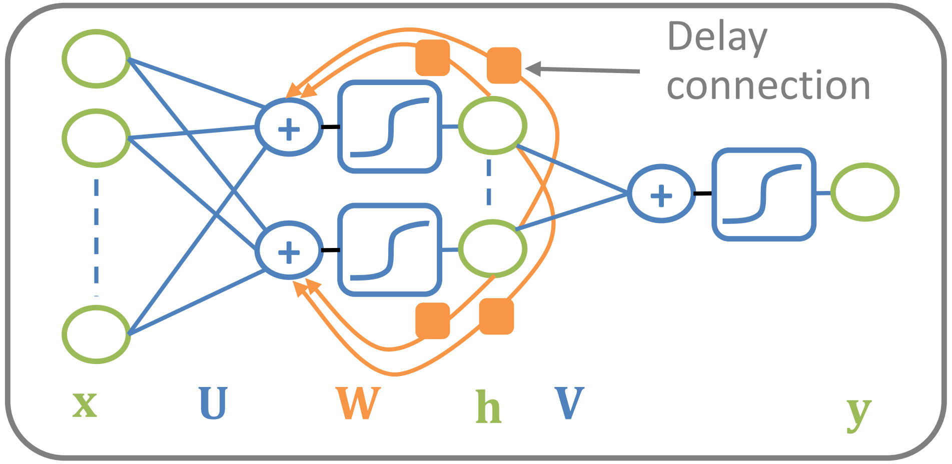
- Deep neural nets suffer from the vanishing/exploiding gradient problem
  - From chain rule $\dfrac{\partial \mathbf{x}_n}{\partial \mathbf{x}_{n-1}} = \mathbf{W}_n^T \mathbf{f}'(\mathbf{z}_n)$ has an important role with many layers

- Convolutional neural nets:
  - Not fully connected nets and weight sharing
  - Rectified linear unit (ReLU) layers

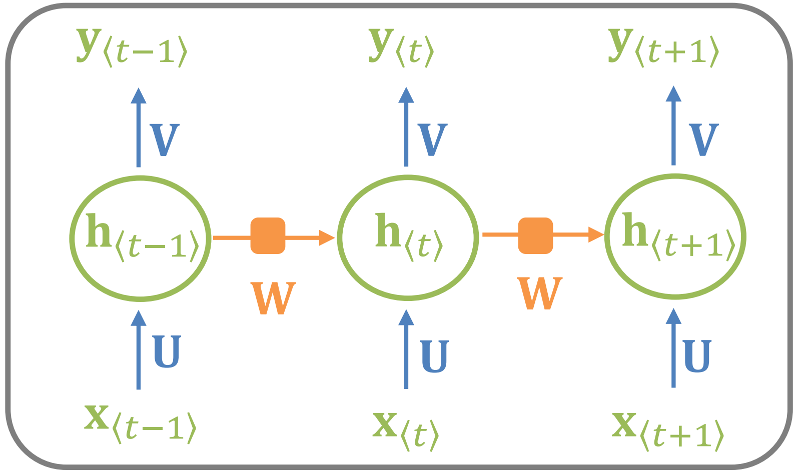




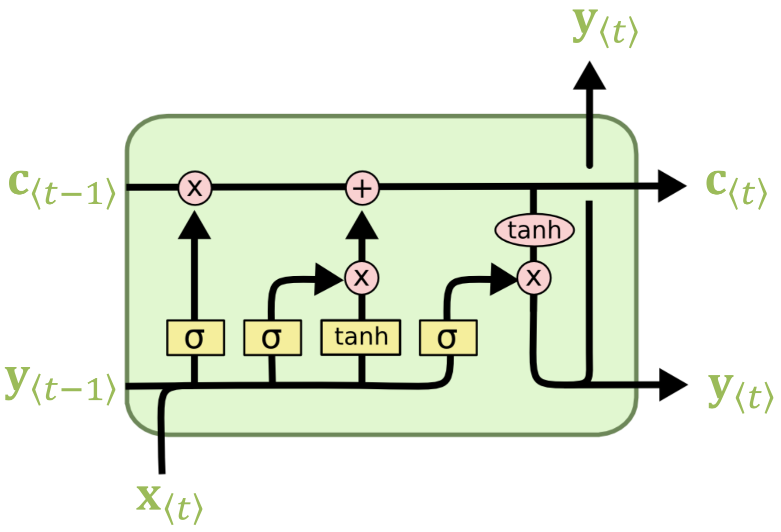
Recurrent neural networks



Unfold



- Backprop through time highly sensible to vanishing/exploiding gradient
- Solutions
  - Truncate backprop:
    - Different time delays
    - Elman network, Jordan networks
  - LSTM: constant error carousel + forget gate



Theories of machine learning

Machine learning fundamentals

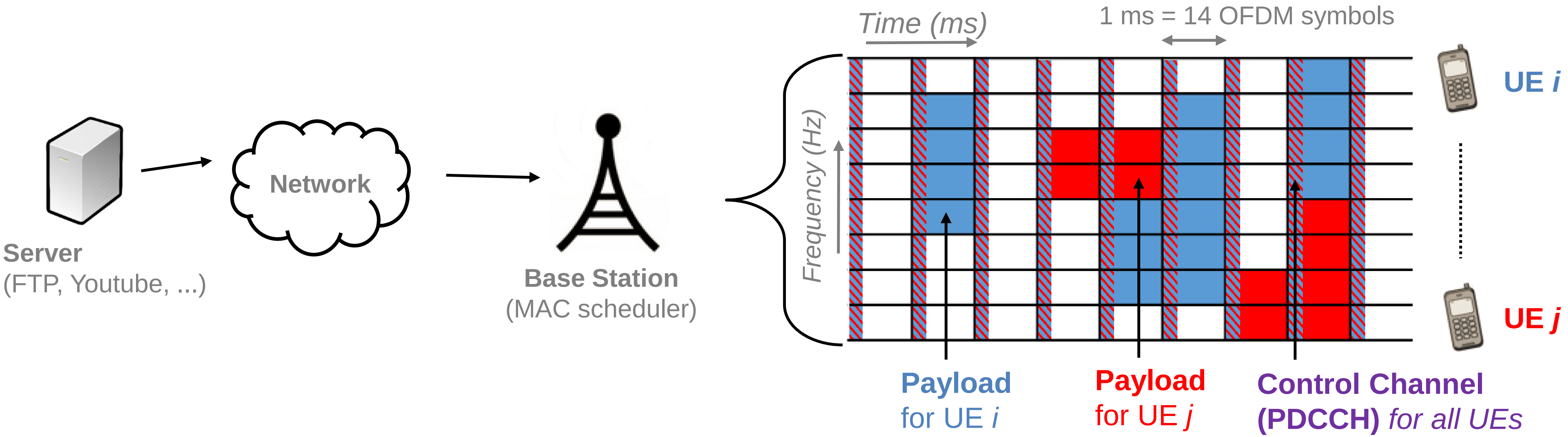
- Statistical learning theory
  - Given the number of samples and hypothesis space, what is the generalization error bound w.r.t. training error ?
- Computational learning theory
  - Given the hypothesis space and the generalization error, how many training samples are required ?
  - Probably approximately correct (PAC) learning algorithm

Agenda

- Machine learning (ML) fundamentals
- ML in practice: Cognitive power control
  - LTE resource allocation and cognitive power control
  - A typical ML workflow and data management
  - Power trajectories and ideal power saving
  - Neural network predictor
  - Reinforcement learning predictor

LTE resource allocation

Machine learning in practice

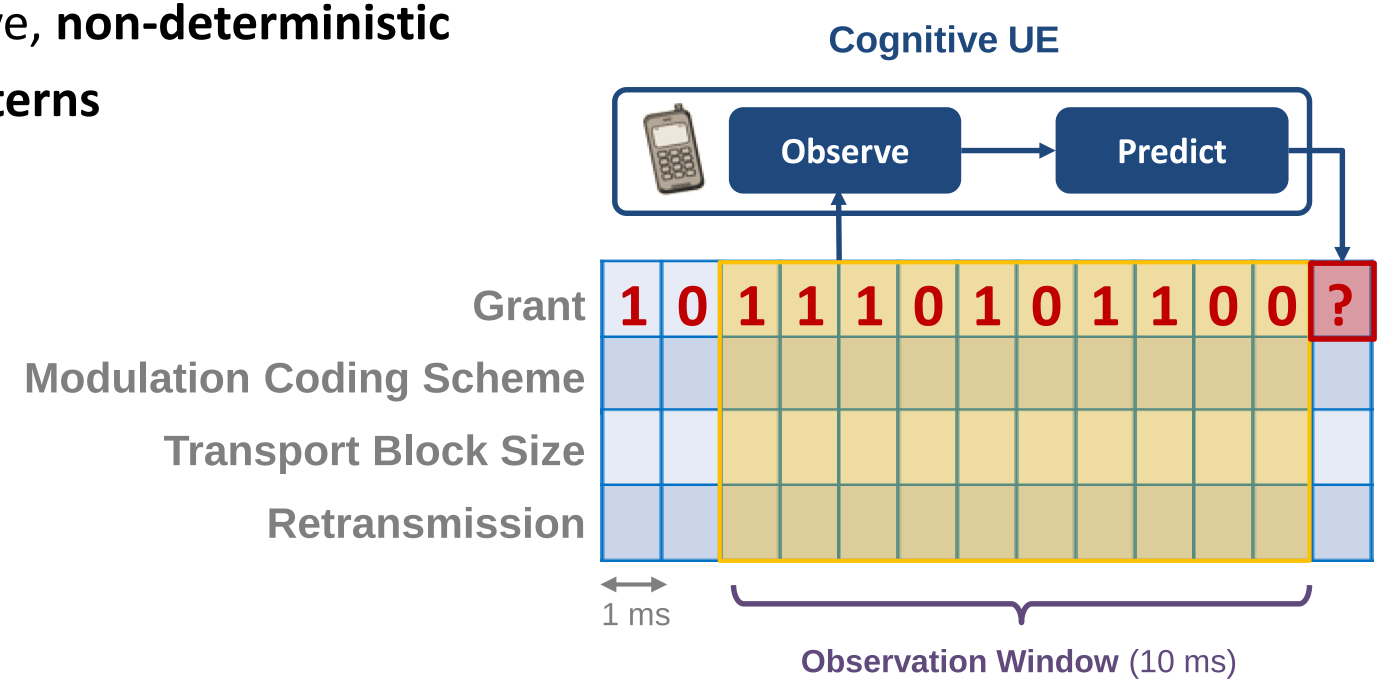


- Every millisecond, the PDCCH should be decoded:
  - **Scenario 1**: The UE has found a grant in the PDCCH and will use it to receive or transmit payload.
  - **Scenario 2:** There is no grant in the PDCCH and power has been used in vain to decode the PDCCH.

Cognitive power control

Machine learning in practice

- If a UE knows in advance that it won‘t receive any grants in the next millisecond, it can **avoid PDCCH decoding**, and therefore save power.
- The base station **MAC scheduler** distributes payload data and grants
  - From UE perspective, **non-deterministic traffic timing patterns**



A typical ML workflow

Machine learning in practice

1. Data collection Streaming data, simulated/live network data, meta-parameter definition and collection, storage.

2. Data preprocessing Efficient data format for queries, split into chunks according markers, format dependent.

3. Feature extraction No differences between formats at the end of this step, need to be able to communicate with experts.

4. Feature preprocessing Data splitting if needed before, normalize and clean features, training set should be obtained.

5. Feature selection Dimensionality reduction algorithms, or automated feature selection via regularization.

6. Model training Choice of the algorithms (supervised learning, reinforcement learning, ...)

7. Model evaluation Choice of the algorithms (supervised learning, reinforcement learning, ...)

Observations:

- Machine learning is inherently an **iterative** exploration
- Efficient **infrastructure** needed (step 1 and 2)
- **Expert knowledge** is mandatory (step 3)
- Always prepare for **scalability** (step 6)
- **Visualize** and analyze samples (step 3, 4 and 7)
- **Manage meta-parameters** (step 1, 2 and 7)

Data management

Machine learning in practice

Home | Dashboard | Examples | Manual | About

Summary

Download | Columns | Select All | Deselect All | Refresh Details

Search:

shapeFile	campTag	lastStartPos	firstStartPos	shapeTag	firstStartTime	colTag	deltaStartTime
log-bb-2018-08-27-09-25-43-000_270-305_1-1	HomeToWork	[48.115750,11.601948]	[48.115752,11.601934]	Open Facebook	1535352847.0	TestApp	18.0
log-bb-2018-08-27-09-25-43-000_362-415_1-1	HomeToWork	[48.104640,11.604992]	[48.104613,11.604928]	Open Mail	1535353155.0	TestApp	33.0
log-bb-2018-08-27-09-25-43-000_469-510_1-1	HomeToWork	[48.094526,11.609309]	[48.094536,11.609312]	Open Google Maps	1535353460.0	TestApp	26.0
log-bb-2018-08-27-09-25-43-000_629-656_1-1	HomeToWork	[48.078419,11.613907]	[48.078525,11.614374]	Open YouTube	1535354154.0	TestApp	29.0
\\imcsmb.imu.intel.com\cogpow\SHAPE_auto\users\jahsue\cont3							
log-bb-2018-08-27-18-29-17-046_25-35_1-1	WorkToHome	[48.078689,11.615514]	[48.078681,11.615551]	FTP DL 5MB	1535384753.0	defaut	4.0
log-bb-2018-08-27-18-29-17-046_222-257_1-1	WorkToHome	[48.108481,11.604270]	[48.108493,11.604319]	Open Messenger	1535385635.0	defaut	30.0
log-bb-2018-08-27-18-29-17-046_259-292_1-1	WorkToHome	[48.108485,11.604284]	[48.108481,11.604272]	Open Google Maps	1535385669.0	defaut	22.0
log-bb-2018-08-27-18-29-17-046_319-386_1-1	WorkToHome	[48.116078,11.602174]	[48.115788,11.602400]	Open Facebook	1535385878.0	defaut	49.0
log-bb-2018-08-27-18-29-17-046_388-422_1-1	WorkToHome	[48.116088,11.602184]	[48.116081,11.602177]	Open Messenger	1535385935.0	defaut	29.0
log-bb-2018-08-27-18-29-17-046_438-461_1-1	WorkToHome	[48.123700,11.597157]	[48.123701,11.597250]	Open Mail	1535386146.0	defaut	18.0
log-bb-2018-08-27-18-29-17-046_487-508_1-1	WorkToHome	[48.126607,11.597099]	[48.126581,11.597077]	Open YouTube	1535386313.0	defaut	15.0
log-bb-2018-08-27-18-29-17-046_585-592_1-1	WorkToHome	[48.133354,11.607773]	[48.133345,11.607768]	FTP DL 5MB	1535386651.0	defaut	4.0
log-bb-2018-08-27-18-29-17-046_608-657_1-1	WorkToHome	[48.133035,11.607926]	[48.133189,11.607845]	Open Messenger	1535386725.0	defaut	59.0
\\imcsmb.imu.intel.com\cogpow\SHAPE_auto\users\jahsue\cont4							
log-bb-2018-08-29-09-50-33-727_784-795_1-1	HomeToWork	[48.078355,11.613734]	[48.078766,11.615019]	Idle screen off	1535528114.0	Defaut	77.0
log-bb-2018-08-29-09-50-33-727_12-22_1-1	HomeToWork	[48.133257,11.607871]	[48.133261,11.607869]	FTP DL 5MB	1535526215.0	Defaut	6.0
log-bb-2018-08-29-09-50-33-727_166-202_1-1	HomeToWork	[48.119594,11.599829]	[48.119585,11.599830]	Open Facebook	1535526810.0	Defaut	26.0
log-bb-2018-08-29-09-50-33-727_271-304_1-1	HomeToWork	[48.108631,11.603942]	[48.108635,11.603918]	Open Messenger	1535527106.0	Defaut	15.0
log-bb-2018-08-29-09-50-33-727_317-367_1-1	HomeToWork	[48.104627,11.604915]	[48.104619,11.604900]	Open Mail	1535527232.0	Defaut	20.0
log-bb-2018-08-29-09-50-33-727_470-533_1-1	HomeToWork	[48.084337,11.606746]	[48.086398,11.606537]	Open Google Maps	1535527716.0	Defaut	54.0
log-bb-2018-08-29-09-50-33-727_557-563_1-1	HomeToWork	[48.082814,11.609064]	[48.082827,11.608660]	Open Podcasts	1535527825.0	Defaut	4.0
log-bb-2018-08-29-09-50-33-727_639-768_1-2	HomeToWork	[48.080149,11.613519]	[48.079911,11.610979]	Open Facebook	1535527917.0	Defaut	51.0
log-bb-2018-08-29-09-50-33-727_639-768_2-2	HomeToWork	[48.080377,11.614595]	[48.080177,11.613613]	Open Facebook	1535527970.0	Defaut	20.0
\\imcsmb.imu.intel.com\cogpow\SHAPE_auto\users\jahsue\cont5							
log-bb-2018-08-29-21-48-12-925_11-21_1-1	Munich	[48.078581,11.616544]	[48.078674,11.615948]	FTP DL 5MB	1535565897.0	WorkToEnglischerGarten	8.0
log-bb-2018-08-29-21-48-12-925_115-152_1-1	Munich	[48.086247,11.606428]	[48.085412,11.606356]	Open Google Maps	1535566310.0	WorkToEnglischerGarten	20.0
log-bb-2018-08-29-21-48-12-925_161-168_1-1	Munich	[48.087184,11.607155]	[48.086978,11.607168]	Open Podcasts	1535566352.0	WorkToEnglischerGarten	4.0
shapeFile	campTag	lastStartPos	firstStartPos	shapeTag	firstStartTime	colTag	deltaStartTime

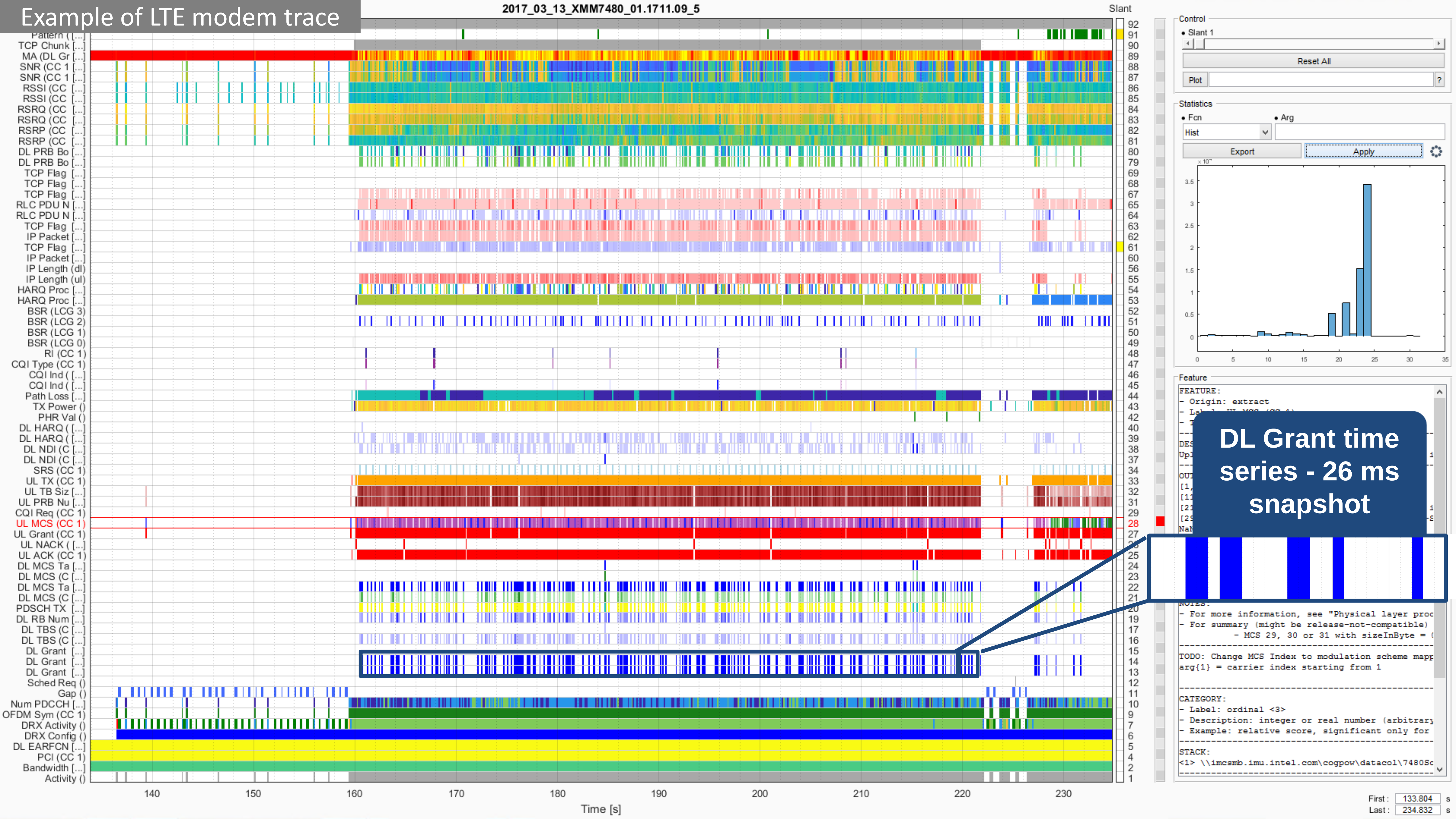
Details

Search:

Name	Value
campMsg	
campTag	WorkToHome
colTag	defaut
deltaStartTime	18.0
device	iPhone-2260904b
firstStartPos	[48.123701,11.597250]
firstStartTime	1535386146.0

Previous 1 2 3 Next

Example of LTE modem trace



Power trajectories

Machine learning in practice

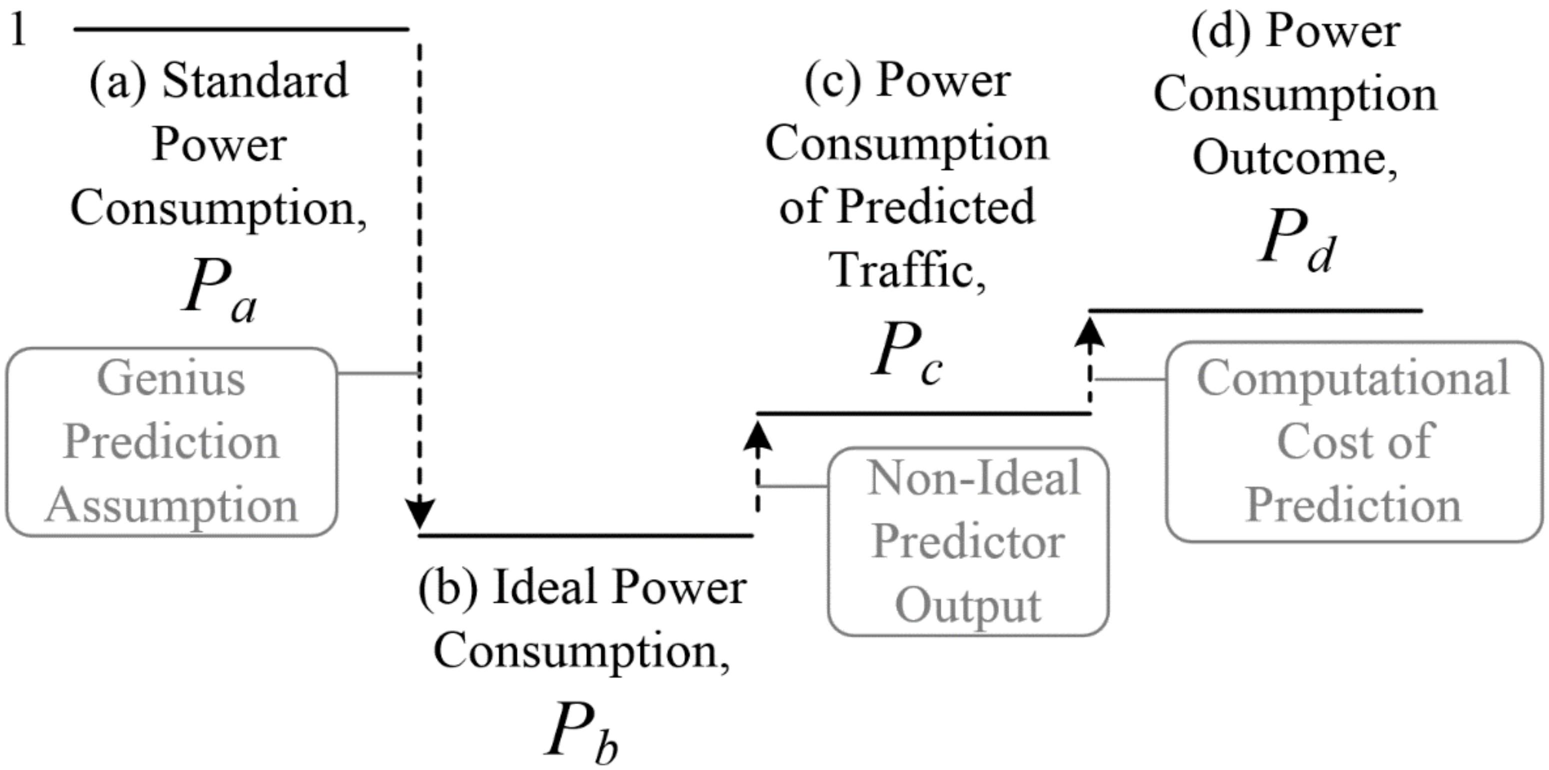
Goal Estimation of the power saving enabled by a ML algorithm at design time without demonstrator. [14]

P_a Power consumption of standard behavior

P_b Power saving potential

P_c Power saving with including prediction errors

P_d Total estimated power saving



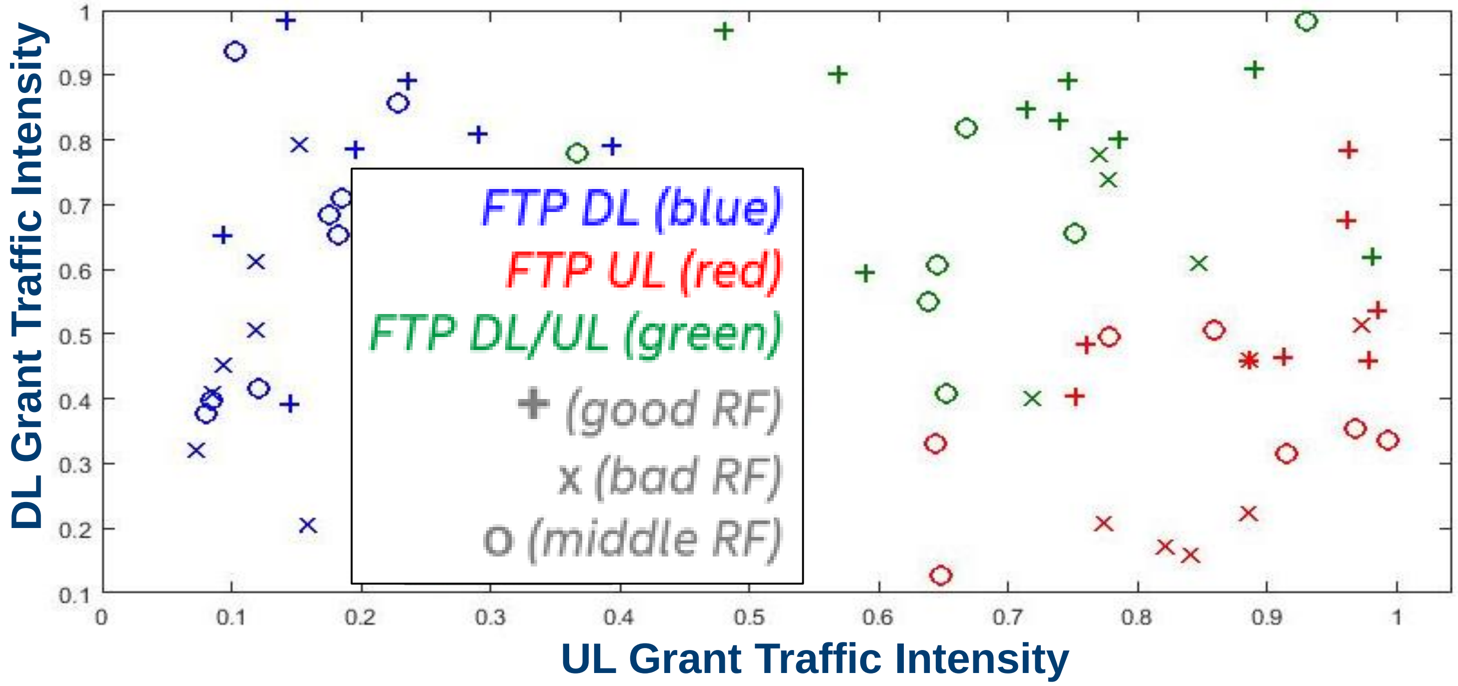
Modem trace data set

Machine learning in practice

Goal Data set from Intel® XMM™ 7480 Modem for LTE-Advanced Services [15] trace server

(6 PB/week; 1 trace ~ 500 MB)

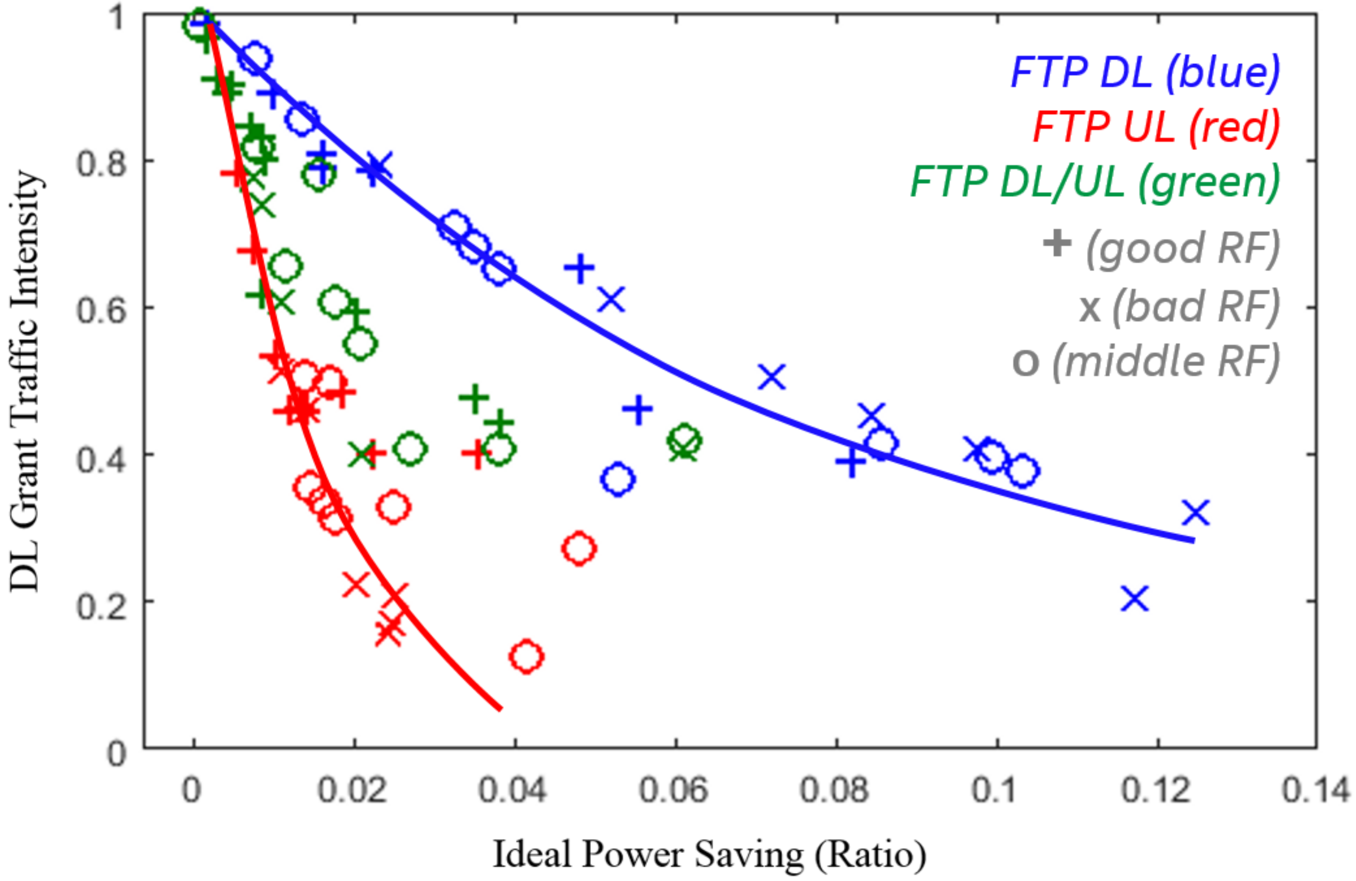
- **Different places and operators**
- **Traffic type** (FTP DL, FTP UL, FTP UL/DL)
- **Radio conditions** (far cell, near cell, middle)
- **Other requirements** (e.g., SW build, CA config)
- **73 traces** selected from ~100000 traces



Ideal power saving

Machine learning in practice

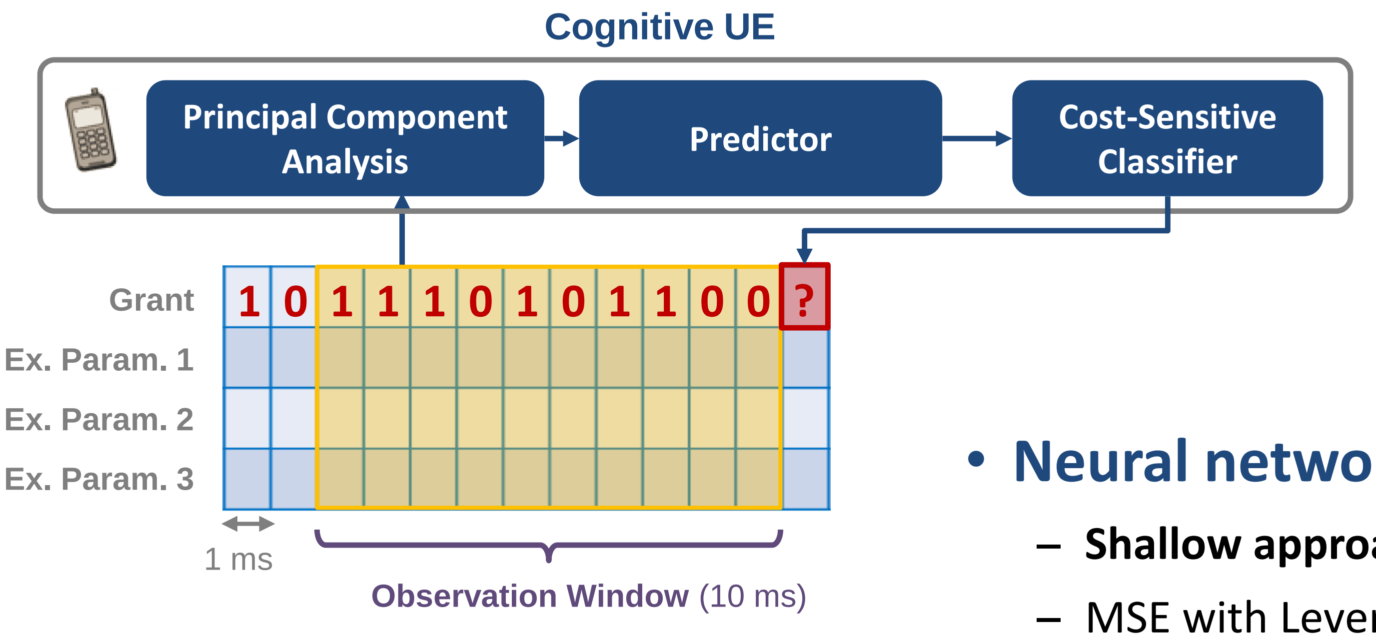
Goal Estimation of the ideal power saving given live network traces assuming genius prediction



- **FTP DL traces are more promising** than FTP UL ones due to the large power contribution of UL payload data transmission
- **Bad RF conditions** lead to a more sporadic reception, i.e., more power saving opportunities
- Up to **12% modem power saving** potential by optimizing PDCCH monitoring

Prediction approach

Machine learning in practice



- **Parameter selection**
  - **Relevant parameters to infer scheduling**
    - Modulation coding scheme
    - Number of resource blocks
    - Re-transmission occurences

- **Neural network predictor**
  - **Shallow approach**: 2 hidden layers of 15 and 20 neurons
  - MSE with Levenberg-Marquardt backpropagation
  - **Linear output activation function**: Better separability

- **Cost-sensitive classifier**
  - **Cost imbalance** between false negatives and false positives, i.e., missing a grant implies throughput degradation.
  - **Cost-sensitive classification** uses decision theoritic approach to define a threshold on the neural network output

2% mean FNR

System design

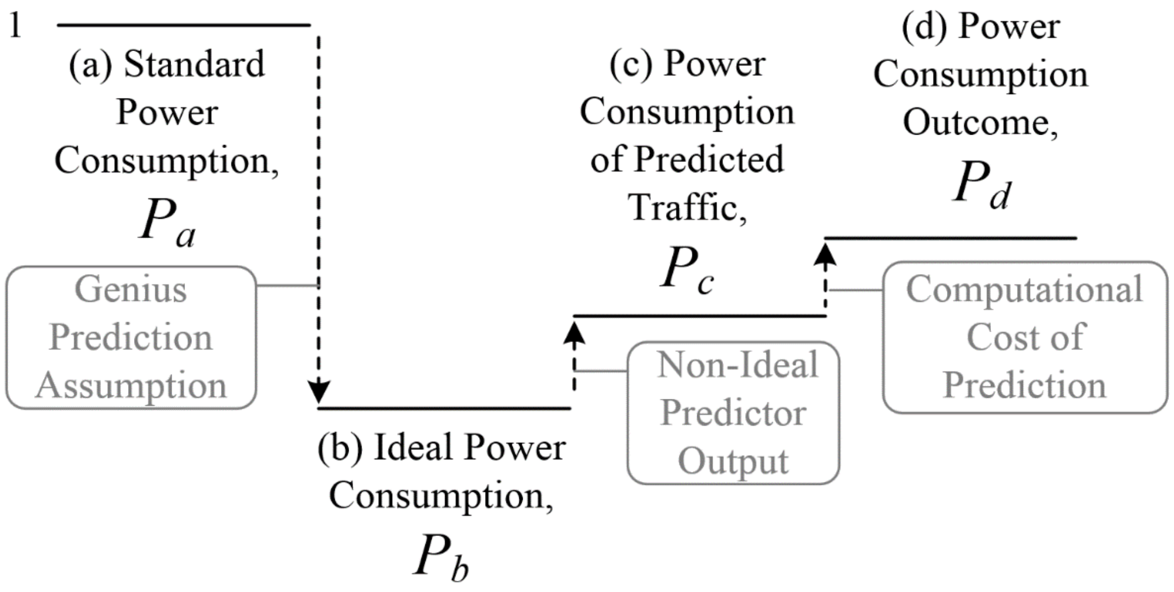
- Computational complexity
  - Typical baseband DSP at **300 MHz**
  - Power consumption of **1 mW/MHz** [1]
  - **No instruction optimizations**: SIMD, vector floating point unit
  - 5 kFLOPs for one prediction: **2 % of a typical DSP time budget**
  - 5 GFLOPs for training: Other approaches should consider the **online/offline training trade-off**

Arithmetic Operation	Complexity
Addition	1 FLOP
Subtraction	1 FLOP
Multiplication	2 FLOPs
Division	4 FLOPs
Exponential	8 FLOPs

- Increase of the classical EDA complexity
  - Area vs. power vs. delay vs. **tolerated error rate** (and its impact on the overall system)
  - Account for the undeterministic nature of such system, assess the **reliability of simulated data**
- Synergies among ML applications
  - Exploitation of the **similarities** between classical machine learning algorithms

Supervised predictor performance

Machine learning in practice



- Main results
  - **12% maximal potential power saving**
  - **2% mean FNR**
  - **2% DSP time budget**
  - **1,7% mean power increase** compared to ideal power consumption
  - **Traffic dependent performance** but promising results for well-defined traffic scenarios

Reinforcement learning approach [16]

Machine learning in practice

- Variable cell behavior:
  - Online training, but high power consumption for NN
- NS3 simulator:
  - No live network testing possible
- Q-learning:
  - Light-weight through tabular representation, e.g. Q-learning

Grant Event s

Downlink Grant
Only Uplink Grant
No Grants

Reward r

Positive:
Expected DL(UL) Grant → Received DL(UL)

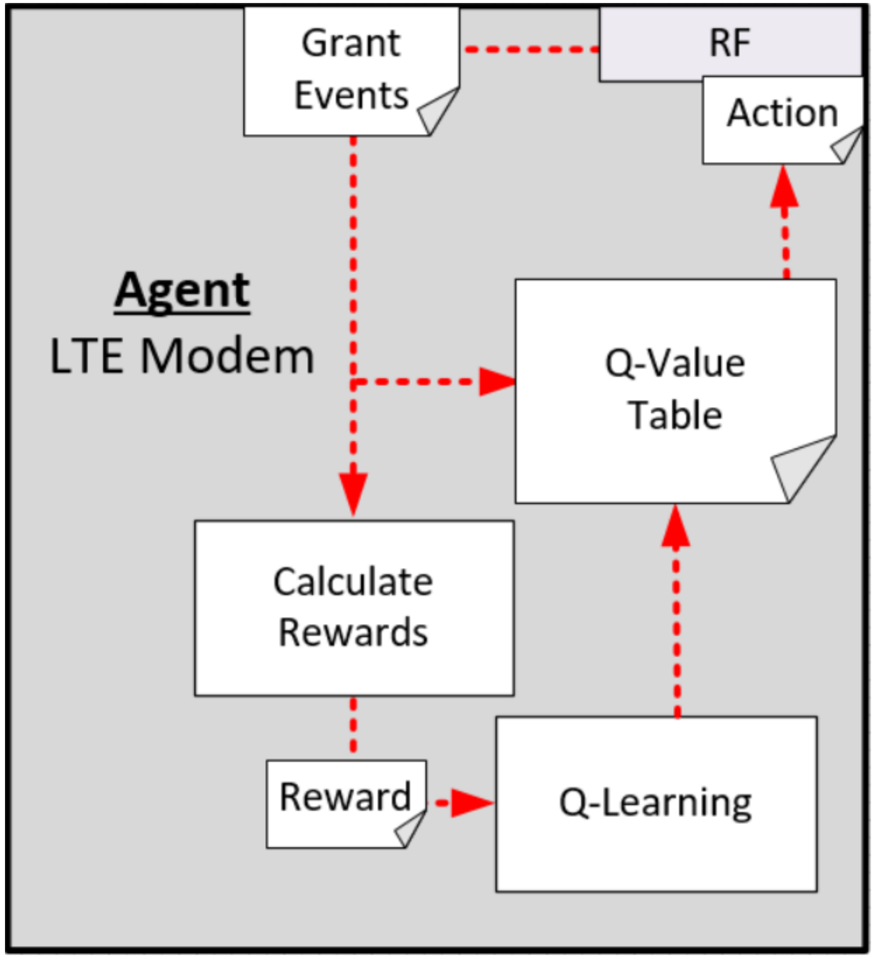
Negative:
- Expected UL Grant → Received DL grant
- Received NACK

Action a

Downlink Grant
Only Uplink Grant
No Grants

Q-learning

Machine learning in practice



Q-Value Q(s,a)

3^{N+1} Entries

For each input(and history) the estimated Long-term reward for each action

Q-Learning

$$Q'(s,a) = (1-\alpha)\cdot Q(s,a) + (\alpha \cdot \max_{a} Q(s',a')\cdot \gamma + r)$$

Conclusion

- Machine learning system
  - Built with **data**, **statistical tools**, robust **workflow** and **expert knowledge**
- Machine learning for power saving
  - **Scenario-specific** trace data collection
  - **Power model** at dedicated abstraction level
  - **Power consumption** estimation of ML algorithms at design time
  - **Power trajectories** for end-to-end power saving estimation
- Cognitive power control outlook
  - Qualify and quantify **network reactions** with network simulator
  - **Online/offline trade-off** through reinforcement learning
  - **Accuracy improvement** with traffic classifier, statistical modeling and LSTM
  - Divide-and-conquer approach with federated learning and trace segmentation

References

[1] McCulloch, W. S., Pitts, W., *A Logical Calculus of the Ideas Immanent in Nervous Activity*, Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943.

[2] Turing, Alan M. "Computing machinery and intelligence." *Parsing the Turing Test*. Springer, Dordrecht, 2009. 23-65.

[3] Moor, James. "The Dartmouth College artificial intelligence conference: The next fifty years." *Ai Magazine* 27.4 (2006): 87.

[4] Rosenblatt, Frank. *Principles of Neurodymanics: Perceptrons and the Theory of Brain Mechanisms*. Spartan books, 1962.

[5] Block, Hans-Dieter. "The perceptron: A model for brain functioning. i." *Reviews of Modern Physics* 34.1 (1962): 123.

[6] Hutchins, John. "The history of machine translation in a nutshell." *Retrieved December* 20 (2005): 2009.

[7] Lighthill, Sir James. "Artificial Intelligence: A General Survey. Part I of Artificial Intelligence': a paper symposium. London: Science Research Council." (1973).

[8] Buchanan, Bruce G., and Edward A. Feigenbaum. "DENDRAL and Meta-DENDRAL: Their applications dimension." *Readings in artificial intelligence*. 1981. 313-322.

[9] Buchanan, Bruce. "Rule based expert systems." *The MYCIN Experiments of the Stanford Heuristic Programming Project*(1984).

[10] LeCun, Yann, et al. "A theoretical framework for back-propagation." *Proceedings of the 1988 connectionist models summer school*. Vol. 1. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

[11] Cohen, Paul R. *Empirical methods for artificial intelligence*. Vol. 139. Cambridge, MA: MIT press, 1995.

[12] Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." (1999): 850-855.

[13] Domingos, Pedro. *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books, 2015.

[14] Ah Sue, Jonathan, et al. "A predictive dynamic power management for LTE-Advanced mobile devices." *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2018.

[15] *Intel® XMM™ 7480 Slim Modem - LTE Advanced*, July 2017, [online] Available: https://www.intel.com/content/www/us/en/wireless-products/mobile-communications/xmm-7480-brief.html.

[16] Brand, Peter, et al. "Reinforcement Learning for Power-Efficient Grant Prediction in LTE." *Proceedings of the 21st International Workshop on Software and Compilers for Embedded Systems*. ACM, 2018.

Questions