Cognition-based networks: applying cognitive science to wireless networking

Andrea Zanella
Cognition-based networks
- Evolution of cognitive communications & networks
- New holistic concept of cognition-based networks
- Use of machine learning tools toward this vision

Examples of application
- QoE-driven video streaming control
- Context-aware handover optimization
Communication systems are more and more complex
- Cognitive (white spaces) transmissions
- Self-organizing networks
- HetNets

All of these are specific cases of a more general approach based on learning and context-awareness
Cognition applied to wireless

- Applying cognition is a way to deal with the complexity and challenges of future systems

- Mitola (2000) and Haykin (2005) actually gave a very general definition of the cognitive paradigm
  - Intelligent observation, learning, decision-making

- Currently, we have only scratched the surface
  - Cognitive networking in a broad sense remains an exciting and largely unexplored research field
Evolution of the cognitive paradigm

- **Cognitive radios**
  - frequency-agile devices for opportunistic access

- **Cognitive radio networks**
  - networks of CRs

- **Cognitive networks**
  - the cognitive approach applied to networking layers and end-to-end

- **Cognition-based networks**
  - drawing from the most recent results in cognitive science
The cognition cycle (simplified)

- Sense
- Act
- Learn
True for humans, true for networks

- **Sense**: nowadays devices are crammed with transducers / sensing apparatuses
  - needs efficient data handling
- **Learn**: optimization algorithms can be run at each node individually
  - needs (i) efficient algos (ii) harmonization
- **Act**: network modifies the environment
  - requires convergence of multiple devices
Supervised vs unsupervised learning

- Supervised learning requires a training set and/or explicit feedback about outcomes

- Unsupervised has no prior knowledge
  - Reveals features that are not predefined, leading to the development of a data-driven worldview
Unsupervised learning

- The way we learn without prior information

- Cognitive stimuli are processed
  - We build a view of the world based on data
  - Generative model: probabilistic view of the world
  - Background for all our cognitive activities

- The worldview provided will be an extremely valuable starting point for goal-specific learning
Cognition-based Network

- Each node of the network:
  - exploits local information to achieve its goal
  - shares (part of) it with its neighbors

- Self-adaptation to the environment to achieve network wide goals

- Cognition applied to the entire network (not just at the PHY and MAC layers)
  - Both vertically and horizontally
Training objective: minimize contrastive divergence (Kullback-Liebler) between input data and (top-down) reconstruction of the data.
Hierarchical processing

A key feature of cortical computation

- Retina
- V1
- V2
- Associative areas
Deep learning

- **Bottom-up:**
  - Higher layer neurons “activate” when input presents some specific features
  - Higher layer provides an abstract representation of input features

- **Top-down:**
  - Activating hidden layer neurons according to their weight and propagating back toward inputs we can generate signals with similar features of training signal

Large scale simulation
Deep network 9 layers (1 billion connections)
10 million (200x200) unlabeled images

(Le et al., 2012)
Properties of Unsupervised learning

- **Generative property**
  - Can be readily used to make predictions about the upcoming input information based on the recent history of the system.
  - It is possible to estimate missing or noisy input terms, and detect anomalies or unexpected patterns in the input signal.

- **Feature extraction**
  - The internal representations extracted by unsupervised learning are generally more informative than those obtained with supervised training.

SIGNET - University of Padova
Properties of Unsupervised learning

- **Compact data representation**
  - Internal representation provides a particular type of coding strategy
- **Synergy with reinforcement learning**
Examples of application

- **Content-based video management**
  - Learn video features and apply this knowledge to some useful networking task

- **Context-dependent handover in HetNets**
  - Learn environmental features and make context-based handover decisions accordingly

- **Prediction of Mobile Devices Discharging Time**
  - Learn the usage pattern and predict the battery charge duration of a smartphone

- ...
Deep Learning for Video Streaming Characterization
Multimedia traffic growth

source: Cisco report (2014)
Mobile video streaming is very demanding:

- High bitrates
  - Compressed HD videos still requires 2-10 Mbit/s

- Low delay
  - Less than 10 ms for interactive videos, less than 250 ms for real-time streaming

- Stable links
  - Link fluctuations are counteracted by a (small) playout buffer → if the buffer empties then the play freezes!

- Content-dependent requirements
  - Quality-rate characteristic depends on the video content
DASH - Dynamic Adaptive Streaming

- **HTTP Standard Server**
- **Client-side Adaptation Engine**
  - The client chooses the proper segments
  - Maximize Quality of Experience (QoE) e.g. avoid rebuffering...
Our approach

- Video Frame sizes
  - Unsupervised deep neural network
  - features
  - (supervised) classifier
  - Rate-distortion characteristics

- DASH request
  - DASH proxy
  - Video quality selection
  - Context-aware resource manager
We consider a test set of 38 video clips, all encoded in an H.264-AVC format.

All the videos are encoded with a 16-frame structure (1 I-frame, 15 P-frames) and compressed with 18 different quantization strategies.

- Transmit rate [bit/s] of video v at compression level c: $r_v(c)$
- Rate Scaling Factor (RSF): $r_v(c) = \log(r_v(c)/r_v(1))$
- Depending on the content, the perceived quality of a given compression level changes

- There are several metrics to measure quality of a video signal

- Here, video quality is expressed in terms of Structural Similarity (SSIM)
Structural Similarity (SSIM)

- SSIM measures the closeness of square sets of pixels, and is computed as

\[
SSIM(X, Y) = \frac{(2\mu_X \mu_Y + c_1)(2\sigma_{XY} + c_2)}{(\mu_X^2 + \mu_Y^2 + c_1)(\sigma_X^2 + \sigma_Y^2 + c_2)}
\]

- Measures image degradation in terms of perceived structural information change
- Represents quality as seen by the human eye
SSIM versus RSF
All the videos exhibit similar trends
- monotonic descent
- a steep “fall” after a threshold

However, there are quantitative differences
- different perceived end quality
- different resource requirements

These characteristics are roughly consistent within the same (homogeneous) video
We introduce a polynomial approximation to express SSIM behavior.

- This provides a compact representation for use in VAC and RM.

\[ F_v^{(n)}(\rho) \approx 1 + a_{v,1}\rho + a_{v,2}\rho^2 + a_{v,3}\rho^3 + \ldots + a_{v,n}\rho^n \]

- A 4-degree polynomial provides a quite accurate approximation of the SSIM vs RSF curve.
Knowledge of these characteristics of the video may be useful for:

(i) QoE-aware admission/congestion control
(ii) determining popular content
(iii) inferring user behavioral characteristics
Proposed approach

Video recognition/classification/SSIM estimation

visible layer: GoP frames’ size

\[ p(h|v) \]

\[ p(v|h) \]
Proposed approach

- Input to the RBM: **frame size only**
  - This is done for a whole GoP (isolated)
  - The RBM “learns” by creating certain patterns in the hidden layer
  - This enables a sparser representation of the input in the hidden layer

- After that, we apply a linear classifier for recognition / classification / SSIM estimation
  - Note: we must train a different linear classifier for each case, while the RBM is trained just once for all
SSIM-based Video Classification
Setting the learning machine

- Dataset is split: Training Set - Test Set
- Input of the RBM (visible layer): 32 units
  - for each of the 16 frames, size of the uncompressed version and of the version compressed at intermediate rate 9
- Hidden layer set (empirically) to 70 units
- The result is compared to just using the raw data as the input of the classifier
Preliminary results

![Bar chart showing raw data for training and test in recognition and classification.](image-url)
Learning via RBM: accuracy

- Root Mean Square Error (RMSE) between exact SSIM and polynomial approximation with estimated coefficients
Exact vs. estimated SSIM curves for two random videos
Cognitive Video Admission Control & Resource Management

- Videos multiplexed into a shared link of capacity $R$
- **Resource Manager (RM)**: detects changes and triggers optimization to adapt video rates to maximize QoE utility function
- **Video Admission Controller (VAC)**: determines whether a new video request can be accepted without decreasing QoE of any video below a threshold $F^*$
At each new video flow request, VAC invokes RM to get the “best” resource allocation according to a specified policy. RM returns the resource that can be assigned to each video. VAC computes the SSIM of each video with the best compression level, compatible with the allotted resources. If estimated SSIM of all active videos is above the quality threshold, the video request is accepted.
The optimization problem addressed by RM is as follows:

$$\Gamma_{opt} = \arg \max_{\Gamma} U(\Gamma, R, \{F_v\}) \quad \text{s.t.} \quad \sum_v \gamma_v \leq 1$$
Possible utility functions

Rate fairness (RF)
\[ U(\Gamma, R, \{F_v\}) = \left[ \sum_v \left| \log \frac{\gamma_v R}{r_v(1)} \right| \right]^{-1} \]

SSIM fairness (SF)
\[ U(\Gamma, R, \{F_v\}) = \min_v F_v(\log(\gamma_v R/r_v(1))) \]

Resource share to be allotted:

RF:
\[ \gamma_v = \frac{r_v(1)}{\sum_j r_j(1)} \]

SF:
\[ \gamma_v = r_v(1) 10^{F_v^{-1}(\phi)} \]

where
\[ \phi = \arg \max \left\{ \sum_v r_v(1) 10^{F_v^{-1}(\phi)} \leq R \right\} \]
Simulation setup

- Poisson video requests (0.66 req./s)
- Average offered load of 11 videos
- Aggregate max rate of $G = 161$ Mbit/s
- $F^* = 0.95$ SSIM value to reach MOS 4

<table>
<thead>
<tr>
<th>SSIM</th>
<th>MOS</th>
<th>Quality</th>
<th>Impairment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\geq 0.99$</td>
<td>5</td>
<td>Excellent</td>
<td>Imperceptible</td>
</tr>
<tr>
<td>[0.95, 0.99)</td>
<td>4</td>
<td>Good</td>
<td>Perceptible but not annoying</td>
</tr>
<tr>
<td>[0.88, 0.95)</td>
<td>3</td>
<td>Fair</td>
<td>Slightly annoying</td>
</tr>
<tr>
<td>[0.5, 0.88)</td>
<td>2</td>
<td>Poor</td>
<td>Annoying</td>
</tr>
<tr>
<td>$&lt; 0.5$</td>
<td>1</td>
<td>Bad</td>
<td>Very annoying</td>
</tr>
</tbody>
</table>
Cognitive video admission control

Link rate over aggregate full-quality video rate
Effect of SSIM approximation

SF RBM-n: SF algorithm, with n-degree polyn. approx. of SSIM curve obtained by RBM approach
Because of SSIM approximation errors, videos may occasionally be accepted even if quality threshold is not met.

Tradeoff on SSIM polynomial approximation: the fewer the coefficients, the coarser the approximation, but the better the RBM coefficient estimate.

Question: is it better to have a well-estimated low degree or a coarsely estimated high-degree polynomial?
- n=2 is too small, n=3 or 4 give very similar results: n=3 is the best choice in this case
QoE-aware proxy vs legacy video clients
Deep Reinforcement Learning for DASH Video Streaming
Reinforcement Learning

At each step t the agent:
- Receives state $s_t$
- Receives reward $r_t$
- Executes action $a_t$

The environment:
- Receives action $a_t$
- Emits state $s_t$
- Emits reward $r_t$

Policy $\pi$ is a behavior function selecting actions given states $a = \pi(s)$

Q-value function $Q^\pi(s, a)$ is expected long-term reward from state $s$ and action $a$ under policy $\pi$

$$Q^\pi(s, a) = \mathbb{E} \left[ r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots | s, a \right]$$
Bellman equation

- Value iteration algorithm
  \[ Q_{i+1}(s, a) = \mathbb{E}_{s'} \left[ r + \gamma \max_{a'} Q_i(s', a') \mid s, a \right] \]

- Optimal value function
  \[ Q^*(s, a) = \mathbb{E}_{s'} \left[ r + \gamma \max_{a'} Q^*(s', a') \mid s, a \right] \]

  Can be approximated by a deep neural network?
  \[ Q(s, a, \theta) \approx Q^\pi(s, a) \]

- Objective function (minimized by Gradient Descent algorithms)
  \[ L(\theta) = \mathbb{E} \left[ \left( r + \gamma \max_{a'} Q(s', a', \theta) - Q(s, a, \theta) \right)^2 \right] \]
Deep Q-Learning

- Find optimal policy from experience:
  - Initially try with random policies
  - Reinforce actions that yield better rewards
  - Progressively move from exploration to exploitation

- Deep Q-learning
  - approximate action-value function $Q(s,q)$ through a neural network
Experience Replay
- Learn from past experience
- Greater data efficiency
- Less correlated training data

Target Network
- More stable training
- Break correlation between target function and Q-network

$$L(\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{U}(R)} \left[ \left( r + \gamma \max_{a'} Q(s', a', \theta^T) - Q(s, a, \theta) \right)^2 \right]$$
**DASH-DQN**

- **State**
  \[ s_t = [q_{t-1}, C_t, D_t, B_t] \]
  - Buffer size
  - Video segment complexity
  - Channel capacity vector
  - Previous quality

- **Quality of Experience (QoE)**
  \[ r(q_t, q_{t-1}, f_t) = q_t - \beta ||q_t - q_{t-1}|| - \lambda f_t \]

- **Structural SIMilarity (SSIM) index**

- **8 Segment Qualities (Possible Actions)**

- **Freezing time**
Deep Q-Network (DQN)

Multi-layer Perceptron with 1 hidden layer: MLP1

Multi-layer Perceptron with 2 hidden layer: MLP2

Long-Short Term Memory: LSTM
Simulation

- Video model: 5 complexities, exponential scene duration
- Capacity model
  - Markov - 500 pre-training videos
  - Real - 40 training videos + 100 test videos
  - NS3 - 40 training videos + 100 test videos
- Comparison
  - Rate-based
  - FESTIVE
  - Q-Learning
Exploration/training/test

Based on synthetic Channel and video traces
Convergence time

![Graph showing convergence time](image-url)
D-Dash fully exploits the buffer to survive connection fluctuations while keeping constant video quality.

Legacy DASH clients try to keep buffer level constant, adapting video quality (worse QoE).
Comparison

Synthetic traces

Real traces
Conclusions

- Optimizing resource allocation for video transmission is challenging
  - many numerical parameters involved
  - subjective QoE issues
  - high signaling exchange

- Learning-based approaches are useful to
  - obtain a compact representation
  - extrapolate the most significant data
  - Provide a framework with no need for prior models


In cellular telephone systems, the term **handover** refers to the process of keeping a mobile active user connected to the BS that offers the strongest radio signal.
Traditional Cellular network
Heterogeneous Networks
Standard 3GPP Handover Procedure

- Long TTT
- Handover Failure
- Short TTT
- Ping-Pong
Objective

- Develop context-aware cognitive handover strategies to maximize the performance of each single mobile user and of the system as a whole in a HetNets scenario.
Steps

① Develop a theoretical model that describes the evolution of the UE state along its trajectory by means of a non homogeneous Markov Chain
② Express the average UE performance as a function of the context parameters
③ Derive a context-aware HO policy (CAHP) that selects the best HO strategy for a UE approaching a femtocell
④ Develop cognitive mechanisms to infer context parameters [still in progress]
System Model

- 1 Master (M-BS) and 1 Femto Base Stations (F-BS), placed at distance $d_{MF}$
- Femtocell coverage area as a circle of radius $R$
- UE constant speed
- Straight path of length $L$
- Uniform incidence angle $\omega$
Propagation & handover models

- RSRP from the h-BS: \( \Gamma_h(a, t) = \Gamma_{h}^{tx} g_h(a) \alpha_h(t) \quad h \in \{M, F\} \)

- Where
  - \( \Gamma_{h}^{tx} \) : h-BS transmit power
  - \( g_h(a) \) : Pathloss gain
  - \( \alpha_h(t) \) : Fast-fading channel gain

- Handover model
  - \( T \) : Time-To-Trigger
  - \( T_{HH} \) : Time for Handover signaling and BS switching
  - \( \gamma_{th} \) : SINR threshold to trigger handover
Sampling at time $T_c >$ channel coherence time $\rightarrow$ independent fading samples $\rightarrow$ average Shannon capacity experienced by UE along the trajectory is

$$
\bar{C}_L = \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} \frac{1}{N_L} \sum_{k=1}^{N_L} \sum_{S \in \{M,F,H\}} \tilde{C}_S(a_k(\omega)) P_S[a_k(\omega)] \, d\omega
$$

where

$$
\tilde{C}_S(a_k(\omega)) = \mathbb{E} \left[ \log_2 (1 + \gamma_S(a_k, kT_c)) \right] = \log_2 (\tilde{\gamma}_S(a_k)) \frac{\tilde{\gamma}_S(a_k)}{\tilde{\gamma}_S(a_k) - 1} \\
\tilde{C}_H(a_k(\omega)) = 0
$$

Average SINR:

$$
\tilde{\gamma}_M(a_k) = \frac{\Gamma_B}{\Gamma_M} g_M(a_k) \quad \tilde{\gamma}_F(a_k) = \frac{1}{\tilde{\gamma}_M(a_k)}
$$
UE’s state: non homogeneous Markov Chain

\[ M_j : \text{connected to the M-BS} \]
\[ F_j : \text{connected to the F-BS} \]
\[ H_j : \text{switching from M-BS to F-BS} \]
\[ \tilde{H}_j : \text{switching from F-BS to M-BS} \]
Transition probabilities

\[ p_{M}^{th}(k) = P[\gamma_M(a_k, kT_c) < \gamma_{th}] = \frac{\gamma_{th}}{\gamma_{th} + \gamma_M(a_k)} \]
\[ p_{F}^{th}(k) = P[\gamma_F(a_k, kT_c) < \gamma_{th}] = \frac{\gamma_{th}}{\gamma_{th} + \gamma_F(a_k)} \]

Transition matrix

\[ \mathbf{P}(k) = \begin{bmatrix} M(k) & V^H_M(k) & \emptyset & \emptyset \\
\emptyset & H(k) & V^F_H(k) & \emptyset \\
\emptyset & \emptyset & F(k) & V^H_F(k) \\
V^M_H(k) & \emptyset & \emptyset & H(k) \end{bmatrix} \]

State probability vector

\[ p(k) = p(0) \prod_{i=0}^{k-1} P(i) \quad p(0) = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \]

UE state probabilities

\[ P_S[a_k(\omega)] = \sum_{i \in \{S_j\}} p_i(k) \quad \text{with} \quad S \in \{M, F, H \cup \tilde{H}\} \]
Accounting for cell load

- HO policy maximizes UE’s estimated average capacity neglecting cells traffic load

- Assuming pilot signals include indication about cell traffic load, we define the load-scaled average capacity of UE in state S as

\[
\overline{C}_S^{\text{load}}(a_k) = \lambda_S \overline{C}_S(a_k) = \lambda_S \log_2(\bar{\gamma}_S(a_k)) \frac{\bar{\gamma}_S(a_k)}{\bar{\gamma}_S(a_k) - 1}
\]

Fraction of available resources of S-BS
Load-aware HO policy

- Keep standard SINR-based HO mechanism
- Set Cell Individual Offset $\lambda_S$ such that relative performance gain is constant:

$$\frac{\bar{C}_M(a_k^*)}{\bar{C}_F(a_k^*)} = \frac{\bar{C}_M^\text{load}(a_k^*)}{\bar{C}_F^\text{load}(a_k^*)}$$

$$\gamma_{th}^{M,\text{load}} = \gamma_{th}^M \frac{\lambda_F}{\lambda_M} \quad \gamma_{th}^{F,\text{load}} = \gamma_{th}^F \frac{\lambda_M}{\lambda_F}$$
### Simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-BS/F-BS Transmit Power</td>
<td>46 dBm/24 dBm</td>
</tr>
<tr>
<td>Distance between BSs</td>
<td>500 m</td>
</tr>
<tr>
<td>Macro/Femto Pathloss exponent</td>
<td>4.5/2.5</td>
</tr>
<tr>
<td>Fading</td>
<td>Rayleigh distribution</td>
</tr>
<tr>
<td>Handover execution time</td>
<td>200 ms</td>
</tr>
<tr>
<td>SINR threshold</td>
<td>0 dB</td>
</tr>
</tbody>
</table>
CAHP: Context-Aware Handover Policy:
- For a certain scenario, and according to the UE speed, it either selects the value of the Time-To-Trigger that gives the maximum average capacity or avoids the HO procedure.

FIX: Fixed Time-To-Trigger Policy:
- A static value $T$ of TTT is used
  - 100 ms, 256 ms, 512 ms
Optimal TTT values

Optimal T values obtained for different speeds and scenarios according to CAHP approach.
Analytical average capacity
Average capacity with different HO policies

![Graph showing average capacity vs. speed for different HO policies](image)
Average capacity

\[ \lambda_M = 0.7 \]
\[ \lambda_F = 1 \]

Non-causal optimal strategy

\[ C_L \text{[bit/s/Hz]} \]

\[ v \text{[Km/h]} \]
Increasing M-BS load

\( \lambda_F = 1 \)

\( v = 60 \text{ Km/h} \)

\( \lambda_M \) vs. \( C_L \) [bit/s/Hz]
Conclusions

- In heterogeneous cellular scenarios, a context-aware Handover policy is beneficial for UEs with respect to conventional Handover management techniques.
- Our optimal context-dependent Handover criterion is based on a solid mathematical analysis and validated by means of simulations.
- Future work:
  - SINR threshold parameter optimization
  - More complex scenarios (multiple BSs and UEs)
  - Machine-learning context estimator
Future work: Neural Networks

- Parameter estimation:
  - UE speed $v$
  - Pathloss exponents
  - Transmit power
  - Distance M-BS - F-BS

- Prediction:
  - Handover? Yes/No

Conclusions

- We have discussed the concept of cognition-based networking:
  - Holistic approach (network-wide, end-to-end, cross-layer)
  - Using the most advanced understanding drawn from cognitive science
  - Machine learning a key ingredient

- Most of the current approaches are too limited and do not address the essence of cognition

- Initial examples show the potential gains

- Although we have been hearing about cognitive radio and networks for years, now it’s time to do it
Several agencies have funded our work in this area, including:

- The European Commission (ARAGON project)
- The US Army Research Office (ARO)
- The CaRiPaRo Foundation in Padova
COMSOC has just started a new journal

IEEE Transactions on Cognitive Communications and Networks
- Started as a JSAC series by Y.C. Liang
- Now greatly expanded in scope

I am Associate Editor, and my colleagues Prof. Michele Zorzi is the EiC of TCCN

Senior advisors: Simon Haykin and Joe Mitola

We have started accepting submissions Jan 21, 2015

Inaugural issue Sep. 2015
Cognition-based networks: applying cognitive science to wireless networking

Andrea Zanella
zanelladei.unipd.it

University of Padova (ITALY)