Deep Neural Networks: Latent Information Extraction, Domain Adaptation & Uncertainty Estimation

Stefanos Kollias$^{1,2}$

$^1$University of Lincoln, UK
$^2$National Technical University of Athens, GR

KSEM 2019, Athens, Greece
28 August 2019
Introduction

In this presentation, based on our recent research on Deep Neural Networks (DNNs) at UoL, we focus on:

• extracting latent information from trained DNNs;
• using this information to visualize/explain the DNN decision making;
• adapting the generated knowledge in other domains;
• providing cues about the decision uncertainty.

Examples from real life problems are used to illustrate the achieved performances.
UK Grand Challenge: Artificial Intelligence & Data

Artificial Intelligence and Machine Learning have started to transform the global economy.

They can be seen as new industries in their own right, whilst transforming business models across sectors.
Industry 4.0 to Society 5.0

Implementation by machines & robots

Data collection by sensors & devices

Analysis by AI, ML

Accumulation as Big Data
Trustworthy AI

- European AI Alliance:
  - Transparent AI
  - Explainable AI
  - Responsible AI
  - Trustful AI
  - Human centric AI
1. Current Developments at UoL

DNNs are developed, analysed & used in

- Industrial Environments: Anomaly Prediction
- Operational Environments: Monitoring and Improving Productivity
- Social Environments: Healthcare, Well being, Culture & Creativity
Industrial Environments

1) Siemens Gas Turbine ML/DL Data Analysis, Fault Prediction - Mindsphere Cloud

2) Nuclear Power Plant DL Signal Analysis, Anomaly Prediction – Simulated & Real Data

(CORTEX, EU H2020 project, 2017-21)
Operational Environments

1) Smart DL/AI Agri-Production Prediction - 1st UK Centre for Agri-Robotics
2) Intelligent Refrigeration Systems - Large Energy Supermarket Savings
3) DL based OCR in Food Packaging - Use By Date and/or Ingredient Verification
4) Environmental Data Analysis - Missing Values Prediction - Water/Rainfall Prediction

(SmartGreen, Interreg project, 2017-21)
Social Environments: Healthcare, Well being, Culture & Creativity

1) Predictive Modelling of Ambulance calls to Care Homes
2) Predicting Parkinson’s Disease from Medical Images
3) Fall Detection of Elderly People Living Alone
4) Video Automated Annotation (Persons, Behaviors)
5) Analysis & Creative Reuse of Cultural Heritage Data

(WeHope, Creative Europe project, 2019-22)
2. DNN Architectures

Various DNN architectures can be used for big data analysis and automatic generation of new features; to name a few:

- Convolutional CNNs
- Recurrent RNNs, C-RNNs, CNN-RNNs
- Generative Adversarial GANs
- Capsule Nets.
Machine /Deep Learning

2012 –
Deep Learning changed the way to deal with analysis of data.

The traditional model of pattern recognition (since the late 50's)
- Fixed/engineered features (or fixed kernel) + trainable classifier

- The model changes with deep learning
  - Trainable features (or kernel) + trainable classifier

End-to-end learning / Feature learning / Deep learning
- Trainable features (or kernel) + trainable classifier

Diagram:
1. Hand-crafted Feature Extractor → “Simple” Trainable Classifier
2. Trainable Feature Extractor → Trainable Classifier
Already Known Architectures (e.g., Neocognitron)

- [Hubel & Wiesel 1962]:
  - **simple cells** detect local features
  - **complex cells** "pool" the outputs of simple cells within a retinotopic neighborhood.

![Diagram showing U_G, U_S1, U_C1, U_S2, U_C2, U_S3, U_C3, U_S4, U_C4, U_M, input layer, contrast extraction, masker layer, recognition layer, "Simple cells", "Complex cells", Multiple convolutions, pooling subsampling]
Automatic feature extraction in Deep Convolutional Neural Networks
Using CNNs for Facial Expression & Behavior Recognition

VA-BATCH

AU-BATCH

EXPR-BATCH

Conv1 → Conv2 → Conv3 → Conv4 → Conv5 → Pool1 → Pool2 → Pool3 → Pool4 → Pool5 → FC → FC

Sigmoid

Valence Arousal

Neutral Angry Disgust Fear Happy Sad Surprise

AU = \{1,2,4,5,6,7,9,10,11,12,15,17,20,23,24,25,26\}
CNN-RNN Architectures (GRU Units)
Various CNN-RNN structures

• Aggregate the outputs of neurons from Fully Connected and/or CNN High/Mid/Low Layers

• Feed one or more RNN Hidden Layers

• Provide final classification or regression through CNN-RNN ensembles
Feature Concatenation in CNN-RNN
Generative Adversarial Networks
StarGANs for Image-Image Translation
DNN Limitations

However, there are problems in DNN use:
- Need of large training datasets (zero/one shot learning is required)
- Lack of transparent & explainable decision making
- Manual annotation of large amounts of data is required, which is practically intractable.

The following can assist in facing these issues:
- Analysis and use of Learnt Structures (Latent Variables).
- Development of self-annotating predictive models.
3. Latent Variable Extraction

1. Extract DNN Learnt Features over Training Data
   \[ \mathcal{R}_p = \{(r(j), j = 1,\ldots,n)\} \]
   \( p \) is the Training Set
   \( r \) denotes the Latent Variables
   \( n \) is the number of Latent Variables

2. Perform Clustering of these Representations
   \[ \hat{T}_{k\text{-means}} = \arg\min_{T} \sum_{j=1}^{k} \sum_{r \in R_p} ||r - \mu_j||^2 \]
   \( \mu \) is the mean value of data in each cluster \( j \)

3. Compute Cluster Centroids & use them to classify Test Data.
Specific Targets

Using the cluster centroids and their annotations we are able to:

• Predict new subjects’ status (through a nearest neighbour criterion); thus explaining why we predict

• Efficiently perform network retraining with new subjects’ data (without forgetting the cluster information)

• Transfer the knowledge learnt by the DNN to other domains/environments (where less input information is available).
An Example: Predicting Parkinson’s

Prediction of Parkinson’s through analysis of medical images (MRI, DaTScans)

A Parkinson’s database comprising 50,000 MRI and 1000 DaTScans from 78 subjects, 55 patients with Parkinson’s & 23 controls has been used (available by permission, from UoL mlearn site).
Predicting with CNN/CNN-RNN (1)

• Through data augmentation more than 100,000 data samples were generated, each including a triplet of (consecutive) MRI and a DaTScan.
• These have been used to train CNN/CNN-RNN architectures to predict the status of the subjects (positive/negative).
• The triple MRI slices and the DaTScan are fed to two independent CNNs, one receiving the MRI triplets as inputs and one receiving the (RGB) DaTSCAN images.
Predicting with CNN/CNN-RNN (2)

• The weights of each CNN were initialised using the ResNet-50 CNN. Their outputs were concatenated at the input of the first FC layer of the CNN & any epidemiological data were also concatenated at this point.

• CNN (with 2 ReLU FC layers) and CNN-RNN (with 1 ReLU layer – GRU 2 layers) have provided best results for Parkinson’s detection (following a detailed ablation study).

• 128-dimensional vectors were extracted from the last GRU layer as latent variables.
Latent Variable Clustering (1)

- Clustering of the 128-dimensional variables representing the classified images in PD/non-PD classes was performed.
- This resulted in identification of 5 clusters, which were then validated by medical experts as referring to:
  - non-patient (non-PD cases)
  - questionable non-PD (edge cases)
  - early-stage PD (stage 1 primarily)
  - standard-stage PD (stage 2)
  - late-stage PD (stage 4).
- Through this latent variable clustering, we increased the learnt DNN ability to perform 2-class classification (PD/non-PD), so as to manage a 5-stage Parkinson’s prediction.
Latent Variable Clustering (2)

A 3-D projection of the 5 cluster centroids
Squares: patients; Stars: non-patients
The DatScans corresponding to the centroids.
4. Retraining DNNs with Annotated Latent Variables (1)

- Whenever new data are collected (e.g., from one or more new subjects) and annotated, they should be used to adapt the knowledge (i.e., the weights) of the trained CNN/CNN-RNN.

- An approach is developed that computes small weight updates (via Taylor expansion) due to the new data, whilst preserving the annotated latent variables (i.e., data corresponding to cluster centroids).
Retraining DNNs with Annotated Latent Variables (2)

• Modify the Minimised Error Criterion, splitting the input data in two sets, the existing input data (Set P) and the new input data (Set P1, also containing the cluster centroids’ inputs):

\[ \mathcal{E} = \mathcal{E}_{P_1} + \lambda \cdot \mathcal{E}_P \]

• Update the CNN/CNN-RNN Weights \( W \)

\[ W' = W + \Delta W \]
When retraining the DNN over the old and new data, we replace the minimised MSE function, over the new data and the latent variable images, with the constraint that:

“DNN outputs and desired outputs are identical”

\[ y'(j) = d(j); \ j = 1, \ldots, s \]
Retraining of DNNs with Annotated Latent Variables (4)

• Retraining is then performed through a constrained error minimization algorithm (such as the gradient projection)

\[ v = V \cdot \Delta W \]

\[ v(j) = d(j) - y(j); \ j = 1, \ldots, s' \]

where V includes weights of the original DNN

• The MSE is minimised over old & new data, but with higher attention to the data corresponding to the latent variables and the new data.
5. DNN Domain Adaptation through Annotated Latent Variables(1)

- In many cases, e.g., general purpose medical centers, DaT Scan equipment may not be available, whilst having access to MRI technology.
- A domain adaptation methodology has been developed, for providing an improved PD prediction in such environments.
- This is achieved by using the annotated latent variables from the combined DaTScan/MRI data analysis, to drive DNN learning from only MRI, towards replication of the latent values as well.
DNN Domain Adaptation through Annotated Latent Variables (2)

• A new DNN loss function is introduced and used, including:
  - a normal MSE minimisation function, over the MRI input data
  \[ \mathcal{E}_1 = \frac{1}{n'} \sum_{j=1}^{n'} (d'(j) - y'(j))^2 \]
  - an additional term which minimises the MSE between the latent variables extracted in the new environment and the original variables extracted in the richer (with DaTScan & MRI medical imaging) environment
DNN Domain Adaptation through Annotated Latent Variables (3)

\[ g(i, j) = u(i) - r'(j), \quad i = 1, \ldots, k; \quad j = 1, \ldots, n' \]

\[ G(i, j) = g(i, j) \cdot (g(i, j))^T \]

\[ \mathcal{E}_2 = \frac{1}{k n'} \sum_{i=1}^{k} \sum_{j=1}^{n'} (z(i, j) - [1 - f(G(i, j))]^2 \]

The training algorithm is adapted to minimise both MSE terms.

\[ \mathcal{E}_{\text{new}} = \eta \mathcal{E}_1 + (1 - \eta) \mathcal{E}_2 \]
Results (1)

**TABLE I: DNN best performing structures on DaT Scan and MRI data**

<table>
<thead>
<tr>
<th>Structure</th>
<th>No FC layers</th>
<th>No Hidden Layers</th>
<th>No Units in FC Layer(s)</th>
<th>No Units in Hidden Layers</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>2</td>
<td>-</td>
<td>2622-1500</td>
<td>-</td>
<td>94%</td>
</tr>
<tr>
<td>CNN-RNN</td>
<td>1</td>
<td>2</td>
<td>1500</td>
<td>128-128</td>
<td>98%</td>
</tr>
</tbody>
</table>

**TABLE II: Training data in each generated cluster**

<table>
<thead>
<tr>
<th>Cluster</th>
<th>No of Data (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t₁</td>
<td>4,3</td>
</tr>
<tr>
<td>t₂</td>
<td>38,4</td>
</tr>
<tr>
<td>t₃</td>
<td>27,6</td>
</tr>
<tr>
<td>t₄</td>
<td>2,3</td>
</tr>
<tr>
<td>t₅</td>
<td>27,4</td>
</tr>
</tbody>
</table>
Results (2)

Performance on 6 test subjects

By retraining the DNN learnt all former & 3 new subjects; also performing slightly better in the 3 rest

TABLE III: Classification of 6 subjects’ data in clusters $t_1$-$t_5$

<table>
<thead>
<tr>
<th>Test case</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
<th>$t_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Patient 1</td>
<td>44</td>
<td>398</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Non Patient 2</td>
<td>10</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Patient 1</td>
<td>3</td>
<td>7</td>
<td>94</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Patient 2</td>
<td>1</td>
<td>7</td>
<td>139</td>
<td>17</td>
<td>20</td>
</tr>
<tr>
<td>Patient 3</td>
<td>3</td>
<td>0</td>
<td>145</td>
<td>18</td>
<td>38</td>
</tr>
<tr>
<td>Patient 4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>72</td>
</tr>
</tbody>
</table>

TABLE IV: Classification of 3 subjects’ data, after retraining, in clusters $t_1$-$t_5$

<table>
<thead>
<tr>
<th>Test case</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
<th>$t_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Patient 1</td>
<td>41</td>
<td>401</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Patient 1</td>
<td>2</td>
<td>5</td>
<td>99</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Patient 4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>73</td>
</tr>
</tbody>
</table>
Results (3)
Domain Adaptation raised accuracy 70.6 to 81.1%

TABLE V: MRI-based Classification of 6 subjects’ data in clusters $t_1$-$t_5$

<table>
<thead>
<tr>
<th>Test case</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
<th>$t_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Patient 1</td>
<td>181</td>
<td>74</td>
<td>179</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Non Patient 2</td>
<td>14</td>
<td>4</td>
<td>44</td>
<td>33</td>
<td>5</td>
</tr>
<tr>
<td>Patient 1</td>
<td>16</td>
<td>0</td>
<td>53</td>
<td>49</td>
<td>2</td>
</tr>
<tr>
<td>Patient 2</td>
<td>6</td>
<td>0</td>
<td>83</td>
<td>80</td>
<td>15</td>
</tr>
<tr>
<td>Patient 3</td>
<td>26</td>
<td>3</td>
<td>130</td>
<td>35</td>
<td>10</td>
</tr>
<tr>
<td>Patient 4</td>
<td>12</td>
<td>0</td>
<td>51</td>
<td>11</td>
<td>6</td>
</tr>
</tbody>
</table>

TABLE VI: MRI-based Classification of 6 subjects’ data, after domain adaptation, in clusters $t_1$-$t_5$

<table>
<thead>
<tr>
<th>Test case</th>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
<th>$t_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non Patient 1</td>
<td>176</td>
<td>147</td>
<td>114</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Non Patient 2</td>
<td>13</td>
<td>41</td>
<td>25</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>Patient 1</td>
<td>13</td>
<td>0</td>
<td>70</td>
<td>35</td>
<td>2</td>
</tr>
<tr>
<td>Patient 2</td>
<td>5</td>
<td>0</td>
<td>116</td>
<td>54</td>
<td>9</td>
</tr>
<tr>
<td>Patient 3</td>
<td>20</td>
<td>2</td>
<td>140</td>
<td>34</td>
<td>8</td>
</tr>
<tr>
<td>Patient 4</td>
<td>9</td>
<td>0</td>
<td>31</td>
<td>5</td>
<td>35</td>
</tr>
</tbody>
</table>
6. Self Annotation through Deep Bayesian Uncertainty Estimation (1)

• We introduce a self-annotating prediction model based on Self-Training of a Bayesian CNN, that leverages variational inference methods of deep models.

• We propose a new inverse uncertainty weighting technique that encourages the Self-training model to learn from more informative data over time, preventing it from becoming lazy by only selecting easy examples to learn from.
Self Annotation through Deep Bayesian Uncertainty Estimation (2)

A real life problem:
Real-time validation of the ‘use-by’ date on food packaging images; unlimited number of images, different contexts, different backgrounds, occlusions, unfeasible annotation
Self Annotation through Deep Bayesian Uncertainty Estimation (3)

- Self-Training is a simple algorithm in which a classifier predicts labels at inference time, and increments the training set with the most confident predictions of the unlabeled data.
- Assuming a Bayesian Neural Network (BNN) formulation, a prior probability distribution $p(W)$ is placed over the set of trainable parameters $W$, with a Gaussian prior distribution being a sensible choice.
- However, the posterior distribution $p(W/X,Y)$ is intractable.
- Dropout is used at test time to perform a Monte Carlo approximation of the posterior distribution of the parameters.
Self Annotation through Deep Bayesian Uncertainty Estimation (4)

- Aleatoric uncertainty relates to sensory noise in the acquisition process of the data; and is therefore inherently irreducible.
- Epistemic uncertainty relates to our uncertainty about the model parameters; it is in fact reducible as we observe more data.
- We target minimising the epistemic uncertainty w.r.t. the variational interpretation of Dropout.
- By penalising very confident output distributions from our BNN we can improve generalization.
Self Annotation through Deep Bayesian Uncertainty Estimation (5)

Experiments with real data (5000 test data):

Normalised confusion matrices of the results obtained from our self-annotation procedure. x and y axes denote the predicted and actual classes, respectively. (a) Refers to the 5000 predicted labels obtained with the lowest prediction uncertainty. (b) Deterministic CNN predicted labels, wherein the thresholds were set based on networks sigmoidal output. (c) Predicted labels from our Bayesian Self-Training approach, trained with a standard binary log-likelihood loss. (d) Similar to (c) but using a Bayesian CNN trained with a penalised binary log-likelihood loss rather than the standard.
Experiments with real data (5000 test data):

### Bayesian CNN (Penalised log-likelihood)

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>#Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT-OK</td>
<td>0.9532</td>
<td>0.9694</td>
<td>0.9612</td>
<td>294</td>
</tr>
<tr>
<td>OK</td>
<td>0.9427</td>
<td>0.9136</td>
<td>0.9279</td>
<td>162</td>
</tr>
<tr>
<td>Avg./Total</td>
<td>0.9494</td>
<td>0.9496</td>
<td>0.9494</td>
<td>456</td>
</tr>
</tbody>
</table>

### Bayesian CNN (Standard log-likelihood)

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>#Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT-OK</td>
<td>0.9679</td>
<td>0.8538</td>
<td>0.9073</td>
<td>212</td>
</tr>
<tr>
<td>OK</td>
<td>0.889</td>
<td>0.9764</td>
<td>0.9306</td>
<td>254</td>
</tr>
<tr>
<td>Avg./Total</td>
<td>0.9248</td>
<td>0.9206</td>
<td>0.9200</td>
<td>466</td>
</tr>
</tbody>
</table>

### Deterministic CNN (Standard log-likelihood)

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>#Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOT-OK</td>
<td>0.9158</td>
<td>0.7682</td>
<td>0.8355</td>
<td>453</td>
</tr>
<tr>
<td>OK</td>
<td>0.7989</td>
<td>0.9287</td>
<td>0.8589</td>
<td>449</td>
</tr>
<tr>
<td>Avg./Total</td>
<td>0.8576</td>
<td>0.8481</td>
<td>0.8472</td>
<td>902</td>
</tr>
</tbody>
</table>
Conclusions

(1) By analysing the knowledge (latent variables) extracted from a trained DNN with 2-class medical image data of Parkinson’s, we were able to reveal a richer facet of this data, through a 5-cluster separation.

(2) By conserving these clusters, we were able to include new knowledge (through retraining with new data) in the DNN, without forgetting its former 5-cluster separation ability.

(3) By reforming the Error minimization function to focus on these clusters, we were able to adapt the learnt knowledge so as to improve DNN performance in less rich input domains.

(4) By Bayesian Self-Annotation we were able to improve prediction of uncertainty in OCV ‘use-by date’ in food packaging images.
Extending the State-of-the-Art

**ML & DNNs**
Combine supervised with unsupervised DNN learning, using representations extracted from trained DNNs:
latent variables, structures, graphs, projections, clusters, auto-encoder patterns.
Use these for transparency & interpretability.

**DNNs & AI**
Combine DNNs with Reasoning Algorithms & semantically update existing Knowledge Bases.


Bibliography

Conferences (2018-2019)

Thank you for your attention