Machine Learning in Agricultural, Environmental and Applied Economics

Hugo Storm, Kathy Baylis, Thomas Heckelei
Outline Part I  ML from an applied econometrics perspective

- Approach to avoid overfitting
- Supervised approaches
  - Shrinkage methods
  - Tree based methods
  - Neural networks
- Unsupervised approaches
- Model complexity versus interpretability
Outline Part II  What ML can add to our toolbox

- Restrictive functional forms
- Extract information from unstructured data
- Large number of explanatory variables
- Causal inference and Identification
- Limitations of simulation models for policy analysis
For today

- Shameless self promotion (✔)
- Quick overview of Neural Networks (NN)
- How can they help?
  - Restrictive functional forms
  - Extract information from unstructured data
  - Deal with a large number of explanatory variables
  - Causal inference and Identification
- Example: NN for policy simulation
Neural networks (NN)

- Can learn highly nonlinear multi-input/output relations
- Highly flexible/nonlinear
- Deep NN have many layers => representation learning
- Less automatic to train than trees but can handle more complex data structures (e.g. Convolutional NN (CNN) for images or Recurrent NN (RNN) for time series/panel data)

$$h_i = \sigma^{[1]}\left( \begin{pmatrix} W^{[1]}_{(4 \times 3)} \end{pmatrix} x_i + b^{[1]}_{(4 \times 1)} \right)$$

$$y_i = \sigma^{[2]}\left( \begin{pmatrix} W^{[2]}_{(2 \times 4)} \end{pmatrix} h_i + b^{[2]}_{(2 \times 1)} \right)$$

Activation functions:
- Relu
- Tanh
Unsupervised approaches

Autoencoder

- Aim to learn joined probability of (x) instead of E(y|x)
- Non-linear version of PCA
- Aim: Extract most relevant features in (x)

Source: https://www.jeremyjordan.me/autoencoders/
How can NNs help?
Many problems are non-linear. Theory provides only weak guidance. Particularly environmental/physical processes. Underfitting can lead to bias. Existing flexible econometric tools struggle with large K and/or N. NN good for time series (instead of AR).
Extract information from unstructured data

- *Unstructured data* = everything that cannot be processed in a spreadsheet
- Examples: (remote sensing) images, text, cell phone records, weather
- Current approach: extract handcrafted features

**ML approaches**
1) End-to-end learning
2) Unsupervised pre-training
3) Transfer learning
4) *Automatic feature creation*
5) *Text methods*
Unstructured data: End-to-end learning

- Raw data as input variables (without preprocessing)
  + No manual feature extraction
  + No information loss
  - Requires large labeled datasets

Rußwurm and Körner (2017)
- Crop classification with remote sensing
  - >137,000 labelled fields, 19 crops
Unstructured data: Unsupervised pre-training

First successful training of deep NN (Hinton et al. 2006)

1) Pretraining NN layers as autoencoder using unlabeled data
2) Train last layer using labeled data

+ Can make use of unlabeled data
- Aims to preserve as much variation as possible (not necessarily the most relevant variation)
Unstructured data: Transfer Learning

- Use models trained in one context as starting point
- E.g. “Representation learning”

Jean et al. (2016) poverty prediction: 
*Transfer learning*
1) VGG trained on ImageNet
2) Predict nightlights intensity classes from daylight images
3) Predict poverty indices

Why does it work? Examples, roofing material and distance to urban areas
Large number of explanatory variables

- In many application $K$ is large and/or high temporal/spatial resolution (scanner/climate date)
- Current approach: Handcrafted collapsing, test driven selection, PCA, Bayesian model selection

**ML approaches**
1) Can better handle large $K$
2) Unsupervised feature extraction
3) RNN/CNN to deal with high temporal/spatial resolution

Source: https://discuss.pytorch.org/t/example-of-many-to-one-lstm/1728
Causal inference and Identification

Causal inference can be thought of as a prediction problem: What would have happened in absence of the treatment?

(i) **Counterfactual simulation** [exogenous treatment]
(ii) Causal forests [selection on observables, heterogeneous effects]
(iii) Double Machine Learning [selection on observables, average effects]
(iv) **ML Panel Methods** [unobserved time-invariant characteristics]
(v) IV with many instruments [endogenous treatment, linear]
(vi) Deep IV estimation [endogenous treatment, nonlinear]
Example: RNN to evaluate farm policy

*How does farm policy affect farm structure?*

Why do we care?
- Much agricultural policy is justified by the ‘need to preserve the family farm’
- Most agricultural policies have differentiated payments based on size and/or payment caps

Challenges:
1. multi-dimensional measure of farm structure
2. non-linear policy (and policy effects) – over multiple outputs
3. spatial dependence
4. dynamics
Why Norway?

- Very detailed data (>70,000 geocoded farms from 1999-present)
- Very complex, activist farm policy, with many kinks
- Regional heterogeneity

Policy notes
Subsidy levels updated each year in negotiation with the farmer’s union
Paid based on last year’s activity level

<table>
<thead>
<tr>
<th>Country</th>
<th>2017</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iceland</td>
<td>55.72</td>
</tr>
<tr>
<td>Korea</td>
<td>53.55</td>
</tr>
<tr>
<td><strong>Norway</strong></td>
<td>52.99</td>
</tr>
<tr>
<td>EU</td>
<td>18.32</td>
</tr>
<tr>
<td>OECD - Total</td>
<td>17.82</td>
</tr>
<tr>
<td>United States</td>
<td>9.88</td>
</tr>
</tbody>
</table>
Policy change

<table>
<thead>
<tr>
<th>Size (head)</th>
<th>2014</th>
<th>2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-50</td>
<td>1326</td>
<td>1000</td>
</tr>
<tr>
<td>51-100</td>
<td>1070</td>
<td>1000</td>
</tr>
<tr>
<td>101-200</td>
<td>347</td>
<td>250</td>
</tr>
<tr>
<td>201-300</td>
<td>210</td>
<td>250</td>
</tr>
<tr>
<td>&gt;300</td>
<td>0</td>
<td>250</td>
</tr>
<tr>
<td>Total cap</td>
<td>280k</td>
<td>560k</td>
</tr>
</tbody>
</table>
Raw data

Marginal Subsidy

Average Subsidy
Recurrent NN with Long Short Term Memory (RNN-LSTM)

- LSTM cells pass information across time in a cell-state vector $c$
- Takes new input each period (X) and use gates to figure out what information it can keep and what we can forget ($a = Gc$)
Intuition

- Cell state encodes past information
- Model learns itself how to encode information
- and which information to keep/forget
  - no need to specify lag structure
  - Lag structure can vary across variables

Example:
- Farm stops milk production
- Specific number of cows every past year might not be relevant
- Sufficient to encode that it had dairy once (maybe the maximum number of cows)
Simulation

<table>
<thead>
<tr>
<th>Size (head)</th>
<th>2014</th>
<th>2015</th>
<th>2016 base</th>
<th>2016 Flat</th>
<th>2016 Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-50</td>
<td>1326</td>
<td>1000</td>
<td>1000</td>
<td>600</td>
<td>1500</td>
</tr>
<tr>
<td>51-100</td>
<td>1070</td>
<td>1000</td>
<td>1000</td>
<td>600</td>
<td>1500</td>
</tr>
<tr>
<td>101-200</td>
<td>347</td>
<td>250</td>
<td>250</td>
<td>600</td>
<td>0</td>
</tr>
<tr>
<td>201-300</td>
<td>210</td>
<td>250</td>
<td>250</td>
<td>600</td>
<td>0</td>
</tr>
<tr>
<td>&gt;300</td>
<td>0</td>
<td>250</td>
<td>250</td>
<td>600</td>
<td>0</td>
</tr>
<tr>
<td>Total cap</td>
<td>280k</td>
<td>560k</td>
<td>560k</td>
<td>560k</td>
<td>560k</td>
</tr>
</tbody>
</table>
Changes in sheep

<table>
<thead>
<tr>
<th>Size (head)</th>
<th>2014</th>
<th>2015</th>
<th>2016 base</th>
<th>2016 Flat</th>
<th>2016 Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-50</td>
<td>1326</td>
<td>1000</td>
<td>1000</td>
<td>600</td>
<td>1500</td>
</tr>
<tr>
<td>51-100</td>
<td>1070</td>
<td>1000</td>
<td>1000</td>
<td>600</td>
<td>1500</td>
</tr>
<tr>
<td>101-200</td>
<td>347</td>
<td>250</td>
<td>250</td>
<td>600</td>
<td>0</td>
</tr>
<tr>
<td>201-300</td>
<td>210</td>
<td>250</td>
<td>250</td>
<td>600</td>
<td>0</td>
</tr>
<tr>
<td>&gt;300</td>
<td>0</td>
<td>250</td>
<td>250</td>
<td>600</td>
<td>0</td>
</tr>
<tr>
<td>Total cap</td>
<td>280k</td>
<td>560k</td>
<td>560k</td>
<td>560k</td>
<td>560k</td>
</tr>
</tbody>
</table>

Flat

Increase
Fodder

<table>
<thead>
<tr>
<th>Size (head)</th>
<th>2014</th>
<th>2015</th>
<th>2016 base</th>
<th>2016 Flat</th>
<th>2016 Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-50</td>
<td>1326</td>
<td>1000</td>
<td>1000</td>
<td>600</td>
<td>1500</td>
</tr>
<tr>
<td>51-100</td>
<td>1070</td>
<td>1000</td>
<td>1000</td>
<td>600</td>
<td>1500</td>
</tr>
<tr>
<td>101-200</td>
<td>347</td>
<td>250</td>
<td>250</td>
<td>600</td>
<td>0</td>
</tr>
<tr>
<td>201-300</td>
<td>210</td>
<td>250</td>
<td>250</td>
<td>600</td>
<td>0</td>
</tr>
<tr>
<td>&gt;300</td>
<td>0</td>
<td>250</td>
<td>250</td>
<td>600</td>
<td>0</td>
</tr>
<tr>
<td>Total cap</td>
<td>280k</td>
<td>560k</td>
<td>560k</td>
<td>560k</td>
<td>560k</td>
</tr>
</tbody>
</table>

**Flat**

![Fodder Flat](image)

**Increase**

![Fodder Increase](image)
Cereals

<table>
<thead>
<tr>
<th>Size (head)</th>
<th>2014</th>
<th>2015</th>
<th>2016 Flat</th>
<th>2016 Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-50</td>
<td>1326</td>
<td>1000</td>
<td>1000</td>
<td>600</td>
</tr>
<tr>
<td>51-100</td>
<td>1070</td>
<td>1000</td>
<td>1000</td>
<td>600</td>
</tr>
<tr>
<td>101-200</td>
<td>347</td>
<td>250</td>
<td>250</td>
<td>600</td>
</tr>
<tr>
<td>201-300</td>
<td>210</td>
<td>250</td>
<td>250</td>
<td>600</td>
</tr>
<tr>
<td>&gt;300</td>
<td>0</td>
<td>250</td>
<td>250</td>
<td>600</td>
</tr>
<tr>
<td>Total cap</td>
<td>280k</td>
<td>560k</td>
<td>560k</td>
<td>560k</td>
</tr>
</tbody>
</table>

Flat

Increase
Implications

- Without explicit economic information, RNN is able to pick up reasonably direct and indirect responses to changes in subsidies
- Able to model more complex dynamic processes than standard AR models
- And more complex spatial patterns
Lots of things yet to do...

- Explore dynamics
- Embed farm fixed effects
- Compare to standard panel model estimation
- Explore how well CNNs can pick up spatial dependence
ML and Environmental Econ: Coming soon

- Thinking hard about selection of input data
- Introducing structure into ML models
- Statistical properties of ML and uncertainty (probabilistic programming)
- Application to Causal Inference