| O6.5 | B0 | B6 | A1 | A5 | F0 | F5 | G0 | G5 | K0 | K5 | M0 | M5 | HD 12993 | HD 158659 | HD 30584 | HD 116608 | HD 9547 | HD 10032 | BD 61 0367 | HD 28099 | HD 70178 | HD 23524 | SAO 76803 | HD 260655 | Yale 1755 |

**Application of Convolutional Neural Network for Stellar Spectral Analysis**

Kaushal Sharma, Inter University Centre for Astronomy and Astrophysics

Collaborators: Ajit Kembhavi, Aniruddha Kembhavi, T. Sivarani, Kaustubh Vaghmare, and Sheelu Abraham
Outline

- Introduction
  - Stellar Spectral Classification
  - Machine Learning/Deep Learning
- Artificial Neural Net
- Convolutional Neural Net
- Autoencoders
- Autoencoders Applications
  - Semi-supervised Classification
  - Outlier Detection
  - Denoising
  - Generative autoencoders
Stellar Spectrum: Distribution of Energy versus wavelength
Stellar Spectrum: Influence of Atmospheric parameters

Model spectra, Coelho (2014)
Stellar Spectrum: Influence of Atmospheric parameters

Model spectra, Coelho (2014)
Stellar Spectrum: Influence of Atmospheric parameters

Model spectra, Coelho (2014)
MK Spectral Classification System

- O (>30,000 K),
- B (10,000 - 30,000 K),
- A (7,500 - 10,000 K),
- F (6000 - 7500 K),
- G (5200 - 6000 K)
- K (3700 - 5200 K) and,
- M (2200 - 3700 K)

- V: dwarfs, log g ~ 4.5 dex
- IV: subgiants, log g ~ 3 dex
- III: giants, log g ~ 1.5 dex
- II: (bright) giants, log g ~ 0.5 dex
- I: supergiants, log g ~ –0.5 dex
Stellar Spectral Classes
Stellar Spectral Classes
Supervised Learning

- Makes machine learn explicitly
- Data with clearly defined output is given
- Direct feedback is given
- Predicts outcome/future
- Resolves classification and regression problems

Unsupervised Learning

- Machine understands the data (Identifies patterns/structures)
- Evaluation is qualitative or indirect
- Does not predict/find anything specific

Reinforcement Learning

- An approach to AI
- Reward based learning
- Learning form +ve & +ve reinforcement
- Machine learns how to act in a certain environment
- To maximize rewards
Classification: Artificial Neural Network

[Diagram of an artificial neural network with layers labeled as input layer, hidden layer 1, hidden layer 2, and output layer.]
Classification: Convolutional Neural Network

Image Credit: http://cs231n.github.io/convolutional-networks/
Input features

- Original Spectrum; HD23511 (F4V)
- Normalized flux at $\lambda=5550$ Å
- Final spectrum between 3600.0-7400.0 Å
- Final spectrum between 3600.0-7400.0 Å
Performance
Performance
## Statistics: Individual Class

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.00</td>
<td>0.88</td>
<td>0.93</td>
<td>88</td>
</tr>
<tr>
<td>B</td>
<td>0.91</td>
<td>0.99</td>
<td>0.95</td>
<td>99</td>
</tr>
<tr>
<td>F</td>
<td>0.96</td>
<td>1.00</td>
<td>0.98</td>
<td>154</td>
</tr>
<tr>
<td>G</td>
<td>0.85</td>
<td>0.94</td>
<td>0.89</td>
<td>251</td>
</tr>
<tr>
<td>K</td>
<td>0.89</td>
<td>0.81</td>
<td>0.85</td>
<td>224</td>
</tr>
<tr>
<td>M</td>
<td>1.00</td>
<td>0.61</td>
<td>0.76</td>
<td>28</td>
</tr>
<tr>
<td>O</td>
<td>0.83</td>
<td>0.83</td>
<td>0.83</td>
<td>6</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.91</td>
<td>0.90</td>
<td>0.90</td>
<td>850</td>
</tr>
</tbody>
</table>

------ Statistics for Test sample (CFLIB)------

Mean difference between   = -0.38  
Absolute Std. Deviation   = 1.72  

Sigma-clipped stats: mean, median, std   = -0.26,  0.00, 1.34
Autoencoders

Encoder

Decoder
Semi-supervised Classification

Unlabelled Data

Labelled Data

Figure adopted from: Bahi M., Batouche M. (2018)
Semi-supervised Classification

![Graph showing predicted class against original class (literature) with SNR color scale.](image)
Semi-supervised Classification

The image shows a confusion matrix with values indicating the recall for each class. The matrix compares the true classes against the predicted classes, with values ranging from 0.00 to 1.00, suggesting a high level of classification accuracy.
## Statistics: Individual Class

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.99</td>
<td>0.77</td>
<td>0.87</td>
<td>88</td>
</tr>
<tr>
<td>B</td>
<td>0.86</td>
<td>0.95</td>
<td>0.90</td>
<td>99</td>
</tr>
<tr>
<td>F</td>
<td>0.93</td>
<td>0.95</td>
<td>0.94</td>
<td>154</td>
</tr>
<tr>
<td>G</td>
<td>0.88</td>
<td>0.88</td>
<td>0.88</td>
<td>251</td>
</tr>
<tr>
<td>K</td>
<td>0.90</td>
<td>0.88</td>
<td>0.89</td>
<td>224</td>
</tr>
<tr>
<td>M</td>
<td>0.88</td>
<td>1.00</td>
<td>0.93</td>
<td>28</td>
</tr>
<tr>
<td>O</td>
<td>0.50</td>
<td>1.00</td>
<td>0.67</td>
<td>6</td>
</tr>
<tr>
<td>avg / total</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>850</td>
</tr>
</tbody>
</table>

------ Statistics for Test sample (CFLIB)------

Mean difference  =  -0.09
Absolute Std. Deviation  =  1.84
Sigma-clipped stats: mean, median, std  =  -0.01,  0.00,  1.27
Pre-trained Model: Architecture

```
input_1: InputLayer
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 576, 1)</td>
<td>(None, 576, 1)</td>
</tr>
</tbody>
</table>

conv1d_1: Conv1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 576, 1)</td>
<td>(None, 576, 64)</td>
</tr>
</tbody>
</table>

max_pooling1d_1: MaxPooling1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 576, 64)</td>
<td>(None, 288, 64)</td>
</tr>
</tbody>
</table>

conv1d_2: Conv1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 288, 64)</td>
<td>(None, 288, 32)</td>
</tr>
</tbody>
</table>

max_pooling1d_2: MaxPooling1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 288, 32)</td>
<td>(None, 144, 32)</td>
</tr>
</tbody>
</table>

conv1d_3: Conv1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 144, 32)</td>
<td>(None, 144, 16)</td>
</tr>
</tbody>
</table>

max_pooling1d_3: MaxPooling1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 144, 16)</td>
<td>(None, 72, 16)</td>
</tr>
</tbody>
</table>

conv1d_4: Conv1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 72, 16)</td>
<td>(None, 72, 16)</td>
</tr>
</tbody>
</table>

up_sampling1d_1: UpSampling1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 72, 16)</td>
<td>(None, 144, 16)</td>
</tr>
</tbody>
</table>

conv1d_5: Conv1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 144, 16)</td>
<td>(None, 144, 32)</td>
</tr>
</tbody>
</table>

up_sampling1d_2: UpSampling1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 144, 32)</td>
<td>(None, 288, 32)</td>
</tr>
</tbody>
</table>

conv1d_6: Conv1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 288, 32)</td>
<td>(None, 288, 64)</td>
</tr>
</tbody>
</table>

up_sampling1d_3: UpSampling1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 288, 64)</td>
<td>(None, 576, 64)</td>
</tr>
</tbody>
</table>

conv1d_7: Conv1D
<table>
<thead>
<tr>
<th>input:</th>
<th>output:</th>
</tr>
</thead>
<tbody>
<tr>
<td>(None, 576, 64)</td>
<td>(None, 576, 1)</td>
</tr>
</tbody>
</table>
```
Classification Model

```
input_1: InputLayer
  input: (None, 576, 1)
  output: (None, 576, 1)

conv1d_1: Conv1D
  input: (None, 576, 1)
  output: (None, 576, 64)

max_pooling1d_1: MaxPooling1D
  input: (None, 576, 64)
  output: (None, 288, 64)

conv1d_2: Conv1D
  input: (None, 288, 64)
  output: (None, 288, 32)

max_pooling1d_2: MaxPooling1D
  input: (None, 288, 32)
  output: (None, 144, 32)

conv1d_3: Conv1D
  input: (None, 144, 32)
  output: (None, 144, 16)

max_pooling1d_3: MaxPooling1D
  input: (None, 144, 16)
  output: (None, 72, 16)

flatten_1: Flatten
  input: (None, 72, 16)
  output: (None, 1152)

dense_1: Dense
  input: (None, 1152)
  output: (None, 512)

dense_2: Dense
  input: (None, 512)
  output: (None, 70)
```
Autoencoders

Applications
- Dimensionality reduction
- Denoising
- Data compression
- Outliers Detection/
  Fraud Detection
Clustering:
PCA versus Autoencoders

Image reconstruction using only first two components
• Signal-to-noise ratio (SNR) greater than 20
• ~60,000 stellar spectra.
Autoencoders

input_1: InputLayer
input: (None, 576, 1)
output: (None, 576, 1)

conv1d_1: Conv1D
input: (None, 576, 1)
output: (None, 576, 64)

max_pooling1d_1: MaxPooling1D
input: (None, 576, 64)
output: (None, 288, 64)

conv1d_2: Conv1D
input: (None, 288, 64)
output: (None, 288, 32)

max_pooling1d_2: MaxPooling1D
input: (None, 288, 32)
output: (None, 144, 32)

conv1d_3: Conv1D
input: (None, 144, 32)
output: (None, 144, 16)

max_pooling1d_3: MaxPooling1D
input: (None, 144, 16)
output: (None, 72, 16)

flatten_1: Flatten
input: (None, 72, 16)
output: (None, 1152)

dense_1: Dense
input: (None, 1152)
output: (None, 512)

dense_2: Dense
input: (None, 512)
output: (None, 70)
Extreme RMSE Outliers

Normalized Flux

Wavelength (Å)
Denoising
Thank you