Music Classification Using Neural Networks

Leo Y. Liu, Lu Wang, Yang Yu

UNC STOR

GTZAN Dataset

- 1000 audio tracks each about 30 seconds long
- 10 types: Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae and Rock
- 22050Hz Mono 16-bit audio files in .wav format

`https://drive.google.com/open?id=0BzPvXAjSgVbXLUxsSWc0c2k1MXM`.

GTZAN Dataset

5 songs are picked form each of the 4 genres: Blues, Classical, Country, Disco.

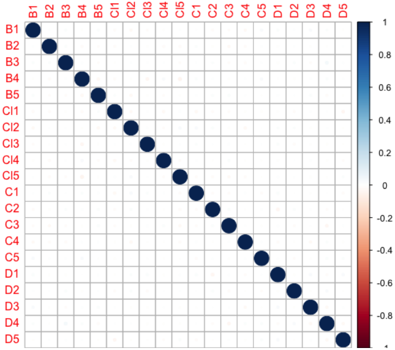


Figure 1: Correlation Plot

GTZAN Dataset

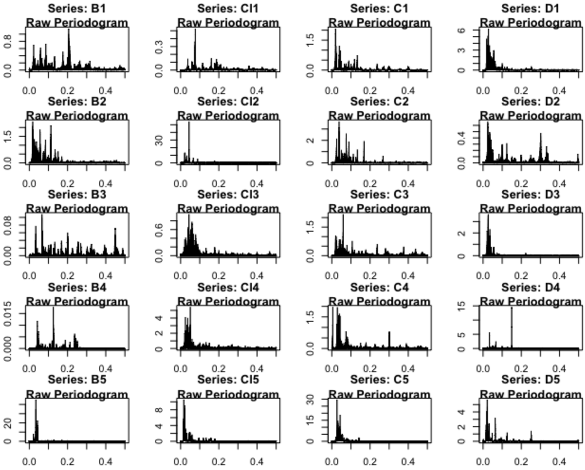


Figure 2: Periodogram

GTZAN Dataset

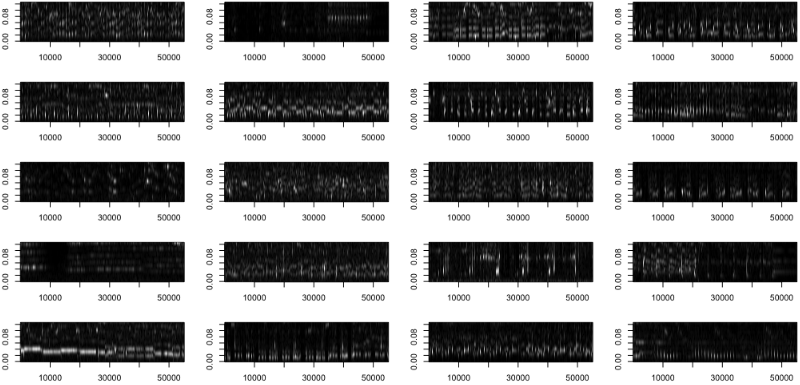
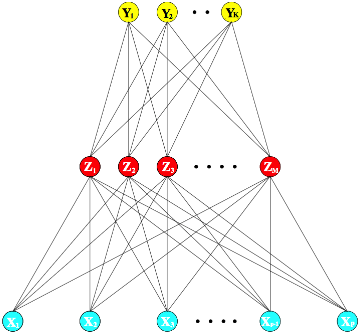


Figure 3: Gabor Transformation

Model: Neural Network

A neural network is a two-stage regression or classification model, typically represented by a network diagram as following.



$$Z_m = \sigma(b_{0m} + w_m^T X), m = 1, \ldots, M,$$

$$T_k = b'_{0k} + \beta_k^T Z, k = 1, \ldots, K,$$

$$Y_k = f_k(X) = g_k(T), k = 1, \ldots, K.$$

Model: Neural Network

- Activation function $\sigma(v)$: sigmoid $\sigma(v) = 1/(1 + e^{-v})$.
- Output function $g_k(T)$: For regression, identity function. In K-class classification, $g_k(T) = e^{T_k} / \sum_{l=1}^{K} e^{T_l}$ (softmax).
- Unknowns: bias and weights $\{b_{0m}, w_m; m = 1, \ldots, M\}$ and $\{b'_{0k}, \beta_k; k = 1, \ldots, K\}$. In total, $M(p+1) + K(M+1)$ unknowns.
- Measure of fit: the sum-of-squared error

$$R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} (Y_k^{(i)} - f_k(X^{(i)}))^2.$$

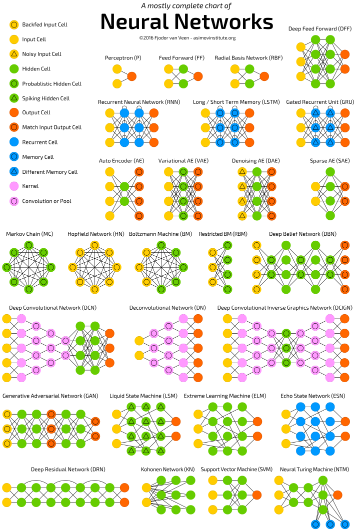
can be used for both regression and classification; the cross-entropy

$$R(\theta) = -\sum_{i=1}^{N} \sum_{k=1}^{K} Y_k^{(i)} \log f_k(X^{(i)})$$

for classification and the corresponding classifier is $G(x) = \text{argmax}_k f_k(X)$.

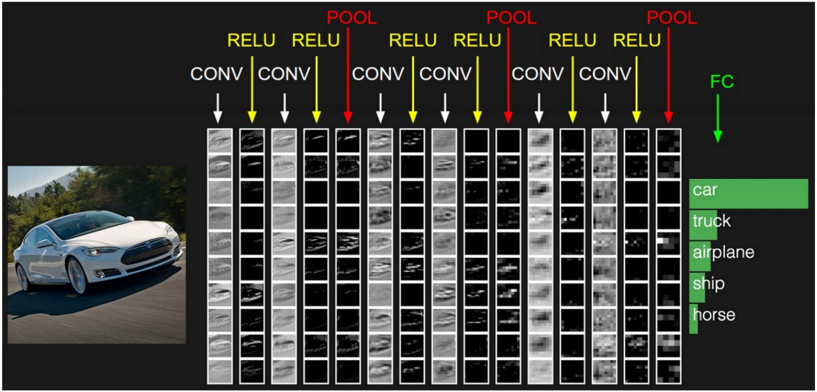
Model: Deep Neural Network / Deep Learning

- Deep neural network/deep learning: a neural network with more than one layers;
- Many variations including recurrent neural network (RNN), auto-encoder (AE), convolutional neural network (CNN);
- The variations are generally modification of the layer structure, activation function and input-output flow.

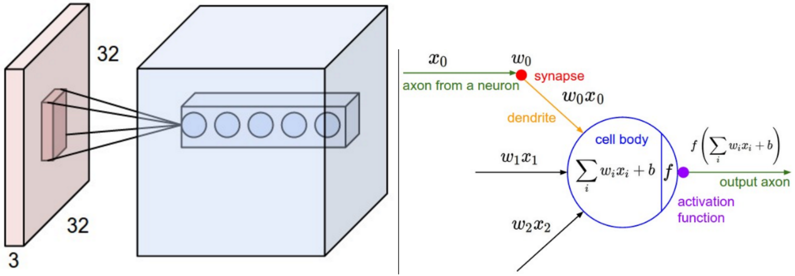


Model: Convolutional Neural Network

It is a deep network with special types of hidden layer: **convolutional layer**, **pooling layer**, and **fully-connected layer** (same hidden layer in regular neural networks).



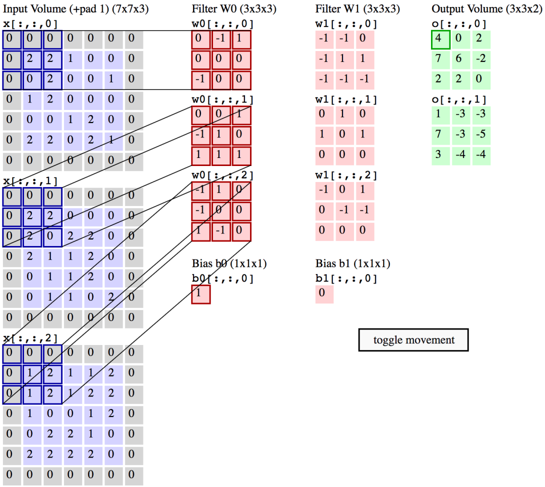
CNN: Convolutional Layer



- Applying the element-wise product between a convolutional kernel (a matrix) and the corresponding regions in the input matrix. Sum them the products up and add a bias term as the input of the next layer.
- Move the kernel along certain direction and with certain stride size.
- Possibly need zero padding.

CNN: Convolutional Layer Continued...

Same weights and bias are used for each of the 3×3 hidden neurons.

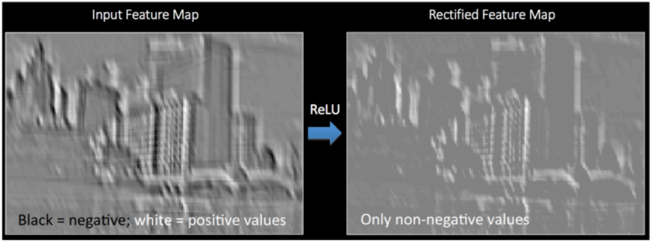


See `http://cs231n.github.io/convolutional-networks/` for an automation illustration.

CNN: ReLU

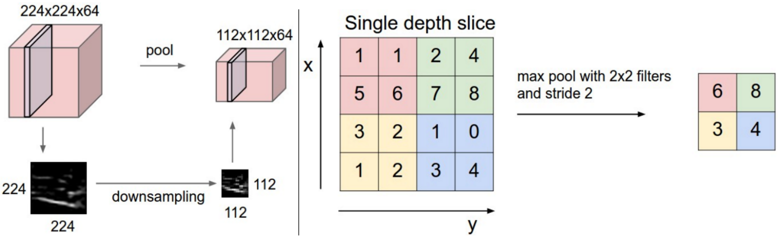
Rectified linear unit:

- Activation layer with $max(0, x)$.
- Sparsity and feature selection.



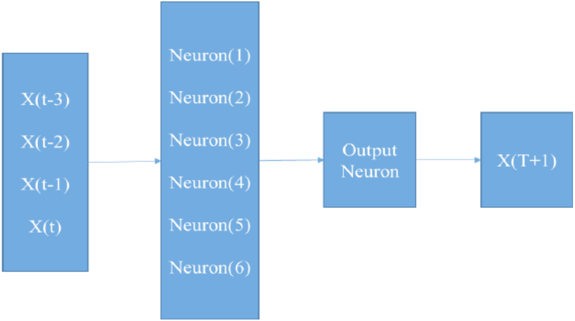
CNN: Pooling Layer

- Down-sample the input layer;
- Max pooling (most popular), average pooling;
- Applying filtering on local regions.

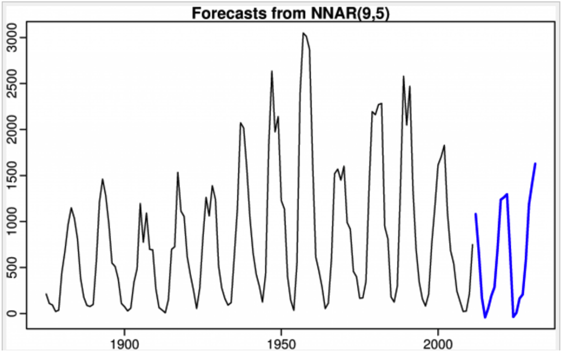


Application: Regression

When time series data shows nonlinearity, we can use neural network to build a neural network autoregression ($NNAR$) instead of AR. An $NNAR(p, K)$ is a neural network with $X_{t-1}, \ldots, X_{t-p}$ as inputs, K neurons in the hidden layer and X_t as the output. Following is an $NNAR(4, 6)$.



Application: Regression



R code

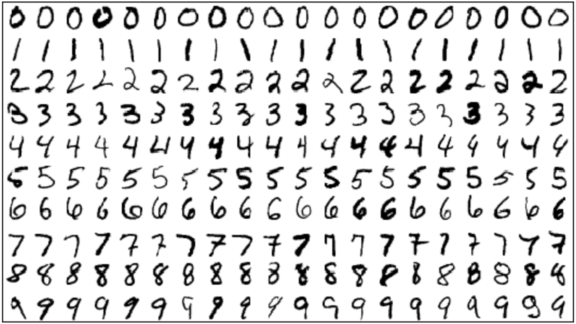
```r
fit <- nnetar(sunspotarea)
plot(forecast(fit,h=20))
```

R code

```r
fit <- nnetar(sunspotarea,lambda=0)
plot(forecast(fit,h=20))
```

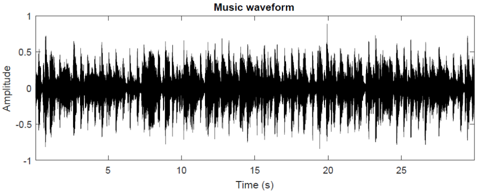
Application: Classification

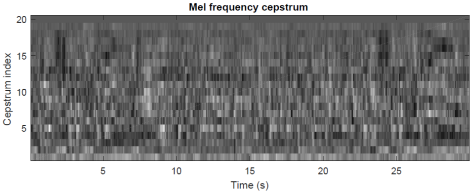
- Handwriting recognition
- Music classification
- so on...



GTZAN Music genres classification

- $n = 1000$;
- $T = 22,050 * 25 = 551,250$;
- $K = 10$;
- Using 80% training 20% testing;
- Preprocessing;
- Classification.

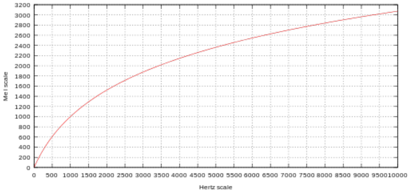




Mel-frequency cepstrum coefficients (MFCC)

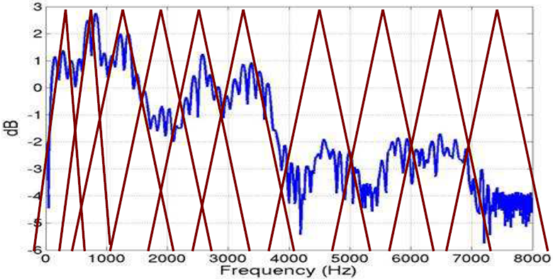
MFCC's characterize the short-term power spectrum of a sound;

1. Take the Fourier transform of a windowed excerpt of a signal.
2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping kernel weights.

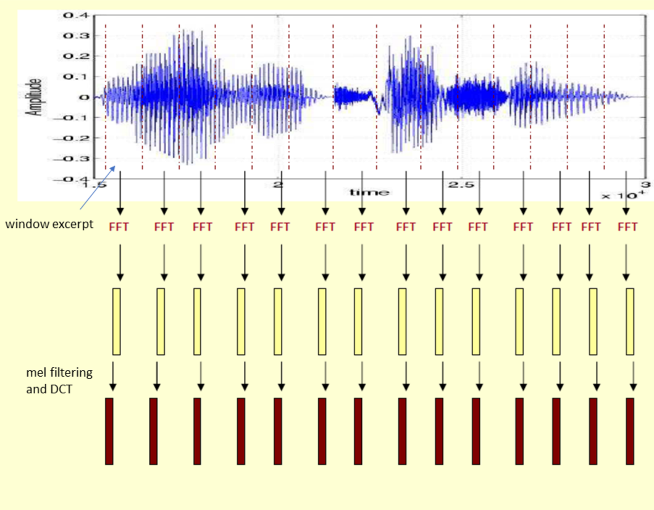


$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) = 1127 \ln\left(1 + \frac{f}{700}\right),$$

3. Take the logs of the powers at each of the mel frequencies.
4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
5. Extract MFCCs as the amplitudes of the resulting spectrum.



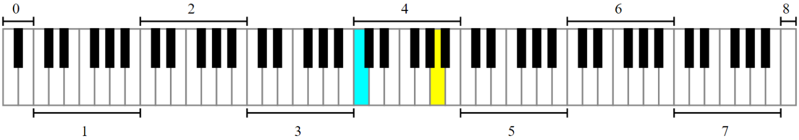
- Apply triangle kernel weight on given frequencies to compute the power spectrum.
- Bandwidth is equal in mel scale, and different in original scale. (small in low frequency and large in high frequency).



Note: window width can be either overlapped or non-overlapped, we used window width of 100 ms with stride size of 25 ms.

Advantages:

- Approximates the human auditory system's response.
  Demo in `http://www.apronus.com/music/flashpiano.htm`
- Downsample the raw data by sampling in the a few frequencies (20hz-8000hz).
  Demo in `https://en.wikipedia.org/wiki/Audio_frequency`
- Utilize the local information, both in time domain and frequency domain.

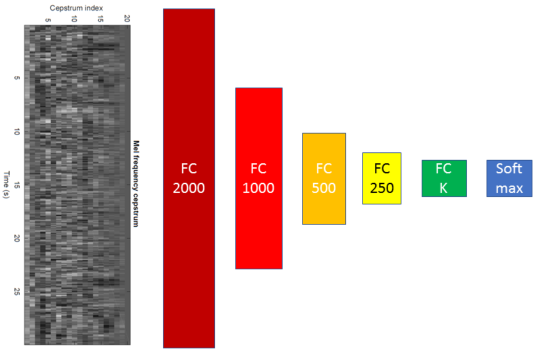


Frequency in hertz
(MIDI note number)

Note \ Octave	−1	0	1	2	3	4	5	6	7	8	9
C	8.176 (0)	16.352 (12)	32.703 (24)	65.406 (36)	130.81 (48)	261.63 (60)	523.25 (72)	1046.5 (84)	2093.0 (96)	4186.0 (108)	8372.0 (120)
C♯/D♭	8.662 (1)	17.324 (13)	34.648 (25)	69.296 (37)	138.59 (49)	277.18 (61)	554.37 (73)	1108.7 (85)	2217.5 (97)	4434.9 (109)	8869.8 (121)
D	9.177 (2)	18.354 (14)	36.708 (26)	73.416 (38)	146.83 (50)	293.66 (62)	587.33 (74)	1174.7 (86)	2349.3 (98)	4698.6 (110)	9397.3 (122)
E♭/D♯	9.723 (3)	19.445 (15)	38.891 (27)	77.782 (39)	155.56 (51)	311.13 (63)	622.25 (75)	1244.5 (87)	2489.0 (99)	4978.0 (111)	9956.1 (123)
E	10.301 (4)	20.602 (16)	41.203 (28)	82.407 (40)	164.81 (52)	329.63 (64)	659.26 (76)	1318.5 (88)	2637.0 (100)	5274.0 (112)	10548.1 (124)
F	10.914 (5)	21.827 (17)	43.654 (29)	87.307 (41)	174.61 (53)	349.23 (65)	698.46 (77)	1396.9 (89)	2793.8 (101)	5587.7 (113)	11175.3 (125)
F♯/G♭	11.563 (6)	23.125 (18)	46.249 (30)	92.499 (42)	185.00 (54)	369.99 (66)	739.99 (78)	1480.0 (90)	2960.0 (102)	5919.9 (114)	11839.8 (126)
G	12.250 (7)	24.500 (19)	48.999 (31)	97.999 (43)	196.00 (55)	392.00 (67)	783.99 (79)	1568.0 (91)	3136.0 (103)	6271.9 (115)	12543.9 (127)
A♭/G♯	12.979 (8)	25.957 (20)	51.913 (32)	103.83 (44)	207.65 (56)	415.30 (68)	830.61 (80)	1661.2 (92)	3322.4 (104)	6644.9 (116)	
A	13.750 (9)	27.500 (21)	55.000 (33)	110.00 (45)	220.00 (57)	440.00 (69)	880.00 (81)	1760.0 (93)	3520.0 (105)	7040.0 (117)	
B♭/A♯	14.568 (10)	29.135 (22)	58.270 (34)	116.54 (46)	233.08 (58)	466.16 (70)	932.33 (82)	1864.7 (94)	3729.3 (106)	7458.6 (118)	
B	15.434 (11)	30.868 (23)	61.735 (35)	123.47 (47)	246.94 (59)	493.88 (71)	987.77 (83)	1975.5 (95)	3951.1 (107)	7902.1 (119)	

Final Model

- Attempted typical CNN, but got disappointing results...
  - Low image features in MFCC's matrix;
  - Algorithm not converged;
  - Li et al. (2010) used 2 hours to training a CNN to classify only 3 genres.
- A deep fully connected CNN, implemented in MATLAB, trained in less than 2 minutes.



Implemented in Matlab. Only a few lines of codes, and less than five minutes of training.

```
layers = [imageInputLayer([21 997 1])
          fullyConnectedLayer(2000)
          fullyConnectedLayer(1000)
          fullyConnectedLayer(500)
          fullyConnectedLayer(250)
          fullyConnectedLayer(n_class)
          softmaxLayer
          classificationLayer()];
```

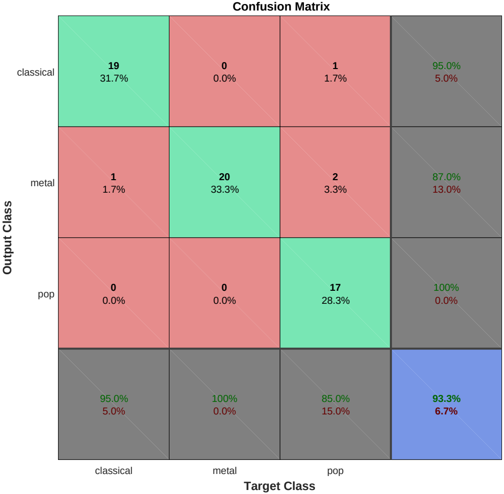
```
Training on single GPU.
Initializing image normalization.
|=========================================================================================|
|     Epoch    |   Iteration  | Time Elapsed |  Mini-batch  |  Mini-batch  | Base Learning|
|              |              |  (seconds)   |     Loss     |   Accuracy   |     Rate     |
|=========================================================================================|
|            1 |            1 |         1.98 |       1.3830 |       28.13% |     1.00e-04 |
|           25 |           50 |         4.05 |       1.2878 |       68.75% |     1.00e-04 |
|           50 |          100 |         5.86 |       1.1332 |       66.41% |     1.00e-04 |
|           75 |          150 |         7.67 |       0.9525 |       60.94% |     1.00e-04 |
|          100 |          200 |         9.47 |       0.8406 |       57.03% |     1.00e-04 |
|          125 |          250 |        11.28 |       0.7804 |       59.38% |     1.00e-04 |
|          150 |          300 |        13.08 |       0.7326 |       66.41% |     1.00e-04 |
|          175 |          350 |        14.90 |       0.6839 |       72.66% |     1.00e-04 |
|          200 |          400 |        16.77 |       0.6305 |       78.91% |     1.00e-04 |
|          225 |          450 |        18.62 |       0.5721 |       82.81% |     1.00e-04 |
|          250 |          500 |        20.42 |       0.5114 |       87.50% |     1.00e-04 |
|          275 |          550 |        22.23 |       0.4514 |       89.06% |     1.00e-04 |
|          300 |          600 |        24.03 |       0.3944 |       91.41% |     1.00e-04 |
```

Binary results

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	100.0%	97.5%	72.5%	82.5%	70.0%	87.5%	80.0%	90.0%	72.5%	72.5%
classical	97.5%	100.0%	92.5%	95.0%	100.0%	87.5%	100.0%	100.0%	97.5%	97.5%
country	72.5%	92.5%	100.0%	82.5%	77.5%	77.5%	95.0%	85.0%	85.0%	65.0%
disco	82.5%	95.0%	82.5%	100.0%	70.0%	95.0%	92.5%	80.0%	72.5%	70.0%
hiphop	70.0%	100.0%	77.5%	70.0%	100.0%	82.5%	90.0%	82.5%	72.5%	65.0%
jazz	87.5%	87.5%	77.5%	95.0%	82.5%	100.0%	97.5%	97.5%	80.0%	87.5%
metal	80.0%	100.0%	95.0%	92.5%	90.0%	97.5%	100.0%	92.5%	100.0%	92.5%
pop	90.0%	100.0%	85.0%	80.0%	82.5%	97.5%	92.5%	100.0%	90.0%	92.5%
reggae	72.5%	97.5%	85.0%	72.5%	72.5%	80.0%	100.0%	90.0%	100.0%	77.5%
rock	72.5%	97.5%	65.0%	70.0%	65.0%	87.5%	92.5%	92.5%	77.5%	100.0%

- Overall above 80%.
- Lower accuracies: country vs. blues (72.5%), hiphop vs. blues (70%), hiphop vs. disco (70%), rock vs. blues (72.5%), country vs. rock (65%), and hiphop vs. rock (65%).
- classical, metal and pop are the three most distinguishable genres;
- blues, country and rock are the three least distinguishable genres.

Multi-category results



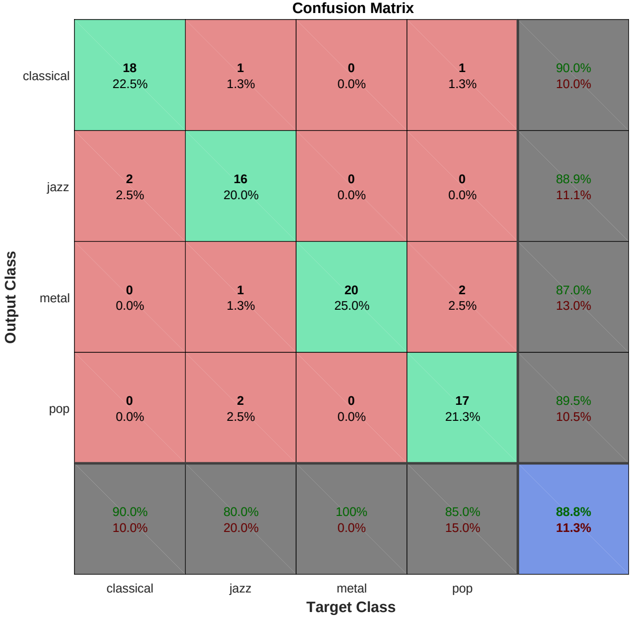

Confusion Matrix
classical
jazz
metal
pop
Output Class
18
22.5%
1
1.3%
0
0.0%
1
1.3%
90.0%
10.0%
2
2.5%
16
20.0%
0
0.0%
0
0.0%
88.9%
11.1%
0
0.0%
1
1.3%
20
25.0%
2
2.5%
87.0%
13.0%
0
0.0%
2
2.5%
0
0.0%
17
21.3%
89.5%
10.5%
90.0%
10.0%
80.0%
20.0%
100%
0.0%
85.0%
15.0%
88.8%
11.3%
classical
jazz
metal
pop
Target Class

Table 1: DAG SVM Results

		Actual			
		Classical	Jazz	Metal	Pop
Predicted	Classical	29	4	1	1
	Jazz	1	20	1	0
	Metal	0	4	26	0
	Pop	0	2	2	29
Accuracy		97%	67%	87%	97%

Table 2: Neural Network Results

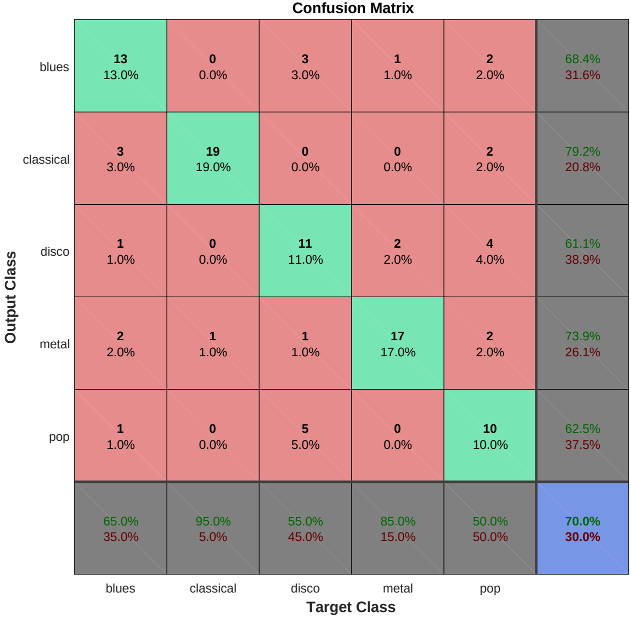
		Actual			
		Classical	Jazz	Metal	Pop
Predicted	Classical	14	0	0	0
	Jazz	1	12	4	0
	Metal	0	0	13	0
	Pop	1	0	0	19
Accuracy		88%	100%	76%	100%

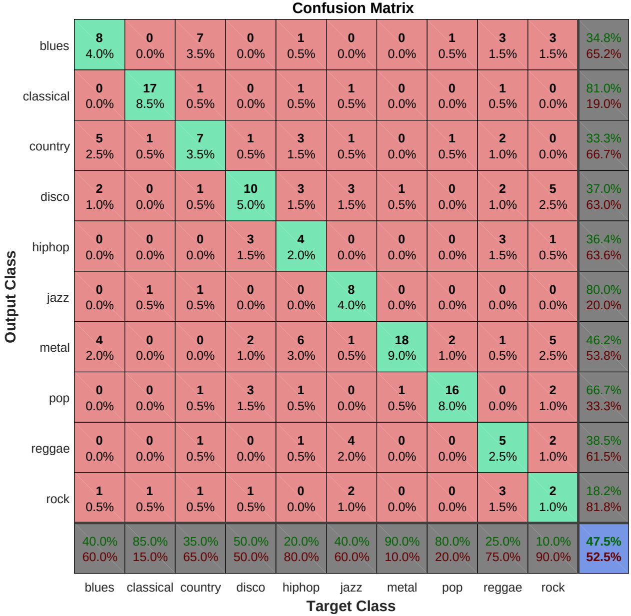
Table 3: k-Means Results

		Actual			
		Classical	Jazz	Metal	Pop
Predicted	Classical	14	16	0	0
	Jazz	2	27	1	0
	Metal	0	0	27	3
	Pop	0	1	1	28
Accuracy		88%	61%	93%	90%

Table 4: k-NN Results

		Actual			
		Classical	Jazz	Metal	Pop
Predicted	Classical	26	9	0	2
	Jazz	4	20	4	1
	Metal	0	1	24	0
	Pop	0	0	2	27
Accuracy		87%	67%	80%	90%


Confusion Matrix
Output Class
blues
classical
disco
metal
pop
Target Class
blues
classical
disco
metal
pop
13
13.0%
0
0.0%
3
3.0%
1
1.0%
2
2.0%
68.4%
31.6%
3
3.0%
19
19.0%
0
0.0%
0
0.0%
2
2.0%
79.2%
20.8%
1
1.0%
0
0.0%
11
11.0%
2
2.0%
4
4.0%
61.1%
38.9%
2
2.0%
1
1.0%
1
1.0%
17
17.0%
2
2.0%
73.9%
26.1%
1
1.0%
0
0.0%
5
5.0%
0
0.0%
10
10.0%
62.5%
37.5%
65.0%
35.0%
95.0%
5.0%
55.0%
45.0%
85.0%
15.0%
50.0%
50.0%
70.0%
30.0%


Confusion Matrix
Output Class
Target Class
blues 8 4.0% 0 0.0% 7 3.5% 0 0.0% 1 0.5% 0 0.0% 0 0.0% 1 0.5% 3 1.5% 3 1.5% 34.8% 65.2%
classical 0 0.0% 17 8.5% 1 0.5% 0 0.0% 1 0.5% 1 0.5% 0 0.0% 0 0.0% 1 0.5% 0 0.0% 81.0% 19.0%
country 5 2.5% 1 0.5% 7 3.5% 1 0.5% 3 1.5% 1 0.5% 0 0.0% 1 0.5% 2 1.0% 0 0.0% 33.3% 66.7%
disco 2 1.0% 0 0.0% 1 0.5% 10 5.0% 3 1.5% 3 1.5% 1 0.5% 0 0.0% 2 1.0% 5 2.5% 37.0% 63.0%
hiphop 0 0.0% 0 0.0% 0 0.0% 3 1.5% 4 2.0% 0 0.0% 0 0.0% 0 0.0% 3 1.5% 1 0.5% 36.4% 63.6%
jazz 0 0.0% 1 0.5% 1 0.5% 0 0.0% 0 0.0% 8 4.0% 0 0.0% 0 0.0% 0 0.0% 0 0.0% 80.0% 20.0%
metal 4 2.0% 0 0.0% 0 0.0% 2 1.0% 6 3.0% 1 0.5% 18 9.0% 2 1.0% 1 0.5% 5 2.5% 46.2% 53.8%
pop 0 0.0% 0 0.0% 1 0.5% 3 1.5% 1 0.5% 0 0.0% 1 0.5% 16 8.0% 0 0.0% 2 1.0% 66.7% 33.3%
reggae 0 0.0% 0 0.0% 1 0.5% 0 0.0% 1 0.5% 4 2.0% 0 0.0% 0 0.0% 5 2.5% 2 1.0% 38.5% 61.5%
rock 1 0.5% 1 0.5% 1 0.5% 1 0.5% 0 0.0% 2 1.0% 0 0.0% 0 0.0% 3 1.5% 2 1.0% 18.2% 81.8%
40.0% 60.0% 85.0% 15.0% 35.0% 65.0% 50.0% 50.0% 20.0% 80.0% 40.0% 60.0% 90.0% 10.0% 80.0% 20.0% 25.0% 75.0% 10.0% 90.0% 47.5% 52.5%
blues classical country disco hiphop jazz metal pop reggae rock

Conclusion

- the deep neural network yields competitive classification accuracy.
- advantage: the prediction power;
- disadvantage: the interpretability;
- the MFCC's capture the key features in the musical audio signals.

Thank You!