Machine learning for music information retrieval

Li Su

Academia Sinica

March 17, 2019

The four elements in a machine learning system

較易取得的 open resources:
metadata, chords, beats

The four elements in a machine learning system

- **Data**: the input
  - Raw data: usually used in image processing
  - Data representation or feature: usually used in audio processing
- **Label**: the goal or answer that you wish the system to learn
  - Categorical label: one datum with one label
  - Attribute label: one datum with multiple labels
- **Model**: define the operation and *metric*
  - Operation: inner product, matrix multiplication, convolution, ...
  - Metric: measure how similar two data representations are
- **Objective function** or *loss function*: measure how good a system learns

In other words: 問題、答案、解題技巧、評分方式

Machine learning problems

Example: genre classification

You have 100 songs, 50 of them are jazz and the others are blues. You want to make your computer able to distinguish between jazz music and blues music.

Feature extraction + classification

- Extract the chroma vectors
- And see if there is a *blue scales* or a *blues chord* progression

Are 'jazz' and 'blues' exclusive?

- Multi-class problems: one data item with exactly one class
  - Categorical loss
- Multi-label problems: one data item with one or more (or no) classes
  - Attribute loss
- In other words: 單選題、多重選擇題

Data and labels

Data representations $\mathbf{x}_i$

- Raw data 波形，罕見
- Spectrogram 主流（librosa 好用）
- Extracted features

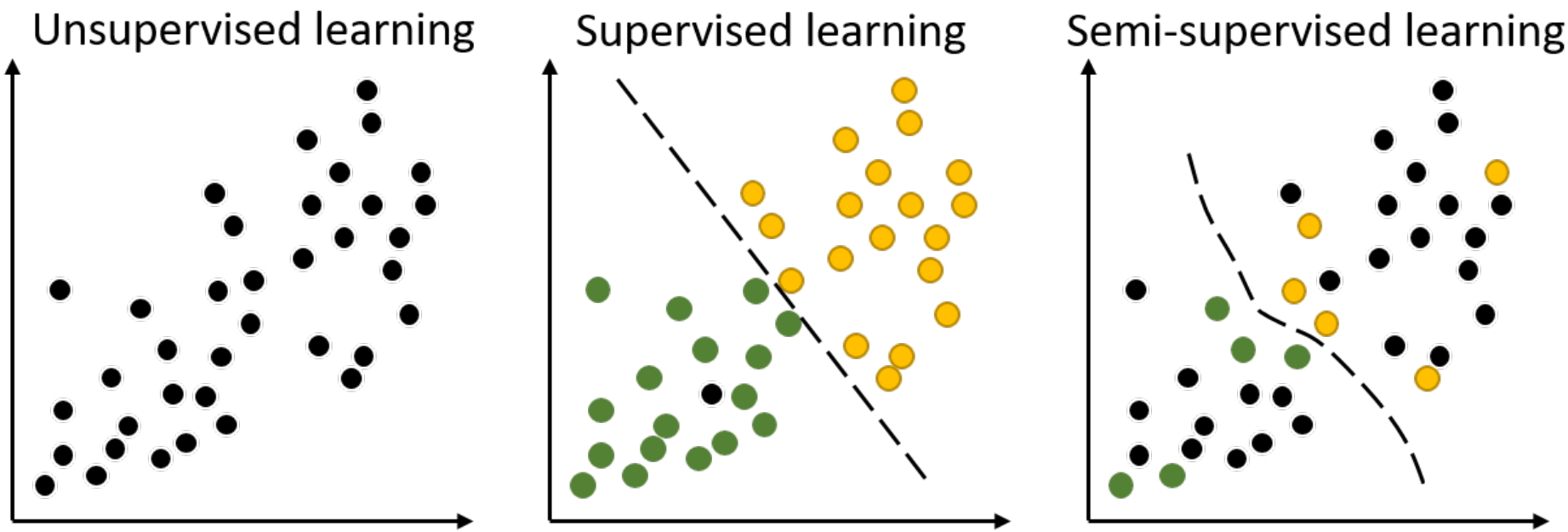
Label representations

- Continuous label $y_i \in \mathbb{R}$ for regression 例：年代
- Discrete label $y_i \in \mathbb{Z}_{\geq 0}$ for multi-class problems 例：曲風 genre
- *One-hot* label $\mathbf{y}_i \in [0,1]^K$ for multi-class problems with K classes
- *One-or-more* label $\mathbf{y}_i \in [0,1]^K$ for multi-label problems with K classes
  a.k.a. multi-hot

Types of machine learning (1)

Unsupervised, supervised, and semi-supervised learning

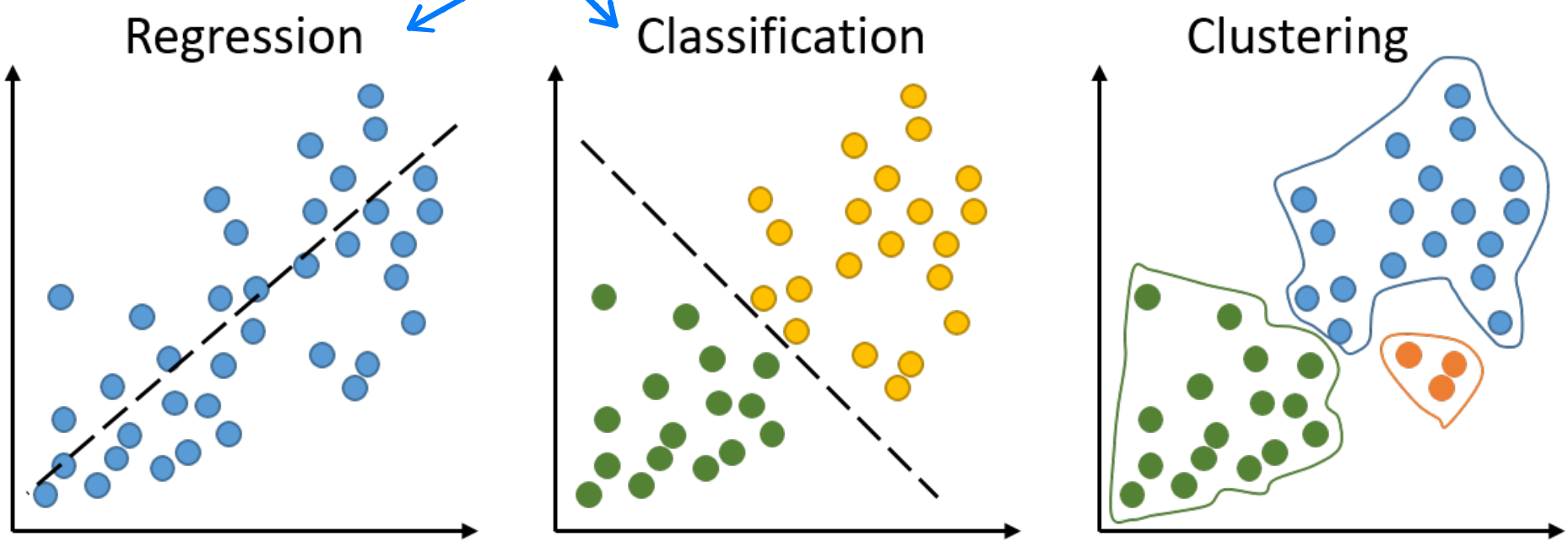
- **Unsupervised learning:** inferring a function to describe the underlying structure of "unlabeled" data
- **Supervised learning:** approximating a mapping function f from input features $\mathbf{x}$ to output labels $\mathbf{y}$; $\mathbf{y} = f(\mathbf{x})$
- **Semi-supervised learning:** approximating a mapping function from the underlying structure of unlabeled data and also a small portion of data are labeled



Types of machine learning (2)

- Regression
- Classification
- Clustering

Music mood 兩法皆有研究



Objectives

Discrete/continuous objectives

	Supervised	Unsupervised
Discrete	**Classification** • Decision trees • Support vector machines • Neural-network classifiers **Regression** • Logistic regression	**Clustering** • k-means • Mean shift • Gaussian mixture models
Continuous	**Regression** • Linear regression	**Dimension reduction** • Principal component analysis • Auto-encoders • Manifold learning

Support vector machines

- Given a training dataset of N items: $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, $\cdots$, $(\mathbf{x}_N, y_N)$
- Or, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]^T$, $\mathbf{y} = [y_1, y_2, \cdots, y_N]^T$
- The *linearly separable* case

$$\text{Minimize } \|\mathbf{w}\| \text{ subject to } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \forall i = 1, 2, \cdots, N \qquad (1)$$

- If data not linearly separable:
- Soft margin
- kernel trick

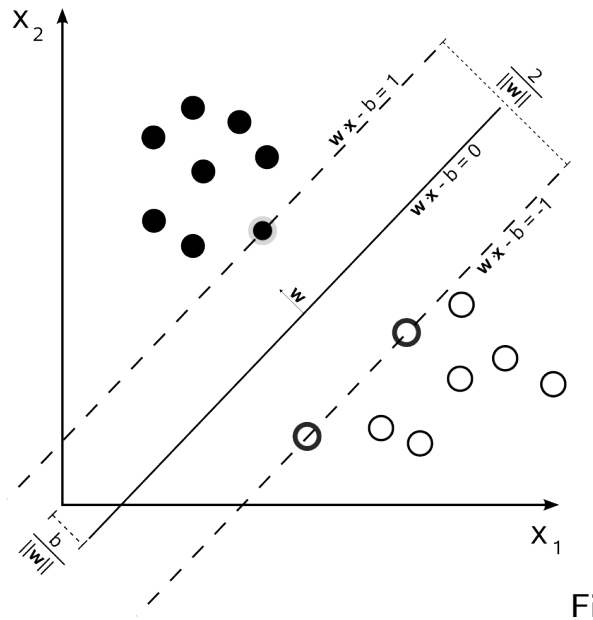


Figure from Wikipedia

Neural networks: the perceptron

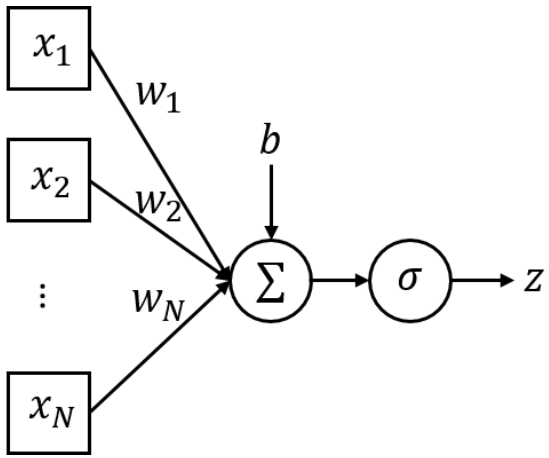
Limitations of a shallow, linear classifier

- Our world is by nature nonlinear
- Many decision processes are of multiple stages
- Solution: nonlinear operation and multi-layered structure

Perceptrons and deep neural networks

- A perceptron: $z = \sigma(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$
- $\sigma(\cdot)$: an element-wise nonlinear function, a.k.a. activation function
- A *fully-connected* (FC) layer (or, a *perceptron*)

$$\mathbf{z} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{2}$$



Activation functions

- Sigmoid
- tanh
- Rectified Linear Unit (ReLU)

$$\sigma_{\text{sigmoid}}(x) = \frac{1}{1+e^{-x}} \tag{3}$$

$$\sigma_{\tanh}(x) = \tanh x \tag{4}$$

$$\sigma_{\text{relu}}(x) = \max(0, x) \tag{5}$$

$$\sigma_{\text{softmax}}(x_j) = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}} \tag{6}$$

Note: for binary (2-class) classification, the sigmoid function is equivalent to the softmax function.

Objective function for a classifier

Given a DNN parametrized by $\mathbf{W} := \{\mathbf{W}^{(k)}\}_{k=1}^{K}$ and $\mathbf{b} := \{\mathbf{b}^{(k)}\}_{k=1}^{K}$ using N training data $\mathbf{x} := x_i{}_{i=1}^{N}$, the network function $\mathbf{z} := f(\mathbf{W}, \mathbf{b}, \mathbf{x})$ is represented as

$$\mathbf{z}^{(0)} = \mathbf{x} \tag{7}$$

$$\mathbf{z}^{(k+1)} = \sigma_{\text{relu}}(\mathbf{W}^{(k)}\mathbf{z}^{(k)} + \mathbf{b}^{(k)}), \quad 2 \leq k \leq K-1 \tag{8}$$

$$\mathbf{z}^{(K)} = \sigma_{\text{softmax}}(\mathbf{W}^{(K-1)}\mathbf{z}^{(K-1)} + \mathbf{b}^{(K-1)}) \tag{9}$$

Given input data $\mathbf{x}_i$, label y_i, network function $f := f(\mathbf{W}, \mathbf{b}, \mathbf{x})$, the optimization problem given by

$$\text{Minimize } \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}(f(\mathbf{W}, \mathbf{b}, \mathbf{x}_i), y_i) \tag{10}$$

How to define the *loss function* $\mathcal{L}$? One of the most commonly used one is the *cross entropy*

Entropy

Definition of entropy

Given N events $(w_1, w_2, \cdots, w_N)$, each of which with probability $(p_1, p_2, \cdots, p_N)$, $\sum_{i=1}^{N} p_i = 1$, $i = 1, 2, \cdots, N$, the entropy $H := H(p_1, p_2, \cdots, p_N)$ is defined as

$$H = -\sum_{i=1}^{N} p_i \ln p_i \tag{11}$$

- Entropy is a measure of 'uncertainty,' 'how hard to predict the outcome,' or 'the amount of surprise' you get when obtaining a sample from a distribution.

Cross-entropy

- Mean square error is the loss function of linear regression, cross-entropy is the loss function of *logistic regression*
- **Binary cross-entropy**: for multi-label problems
- Given the ground truth z and prediction p after the *sigmoid* function

$$L = -z \ln p - (1 - z) \ln(1 - p) \tag{12}$$

- **Categorical cross-entropy**: for multi-class problems
- p are the predictions after the *softmax* function, z are the ground truth
- i denotes the data point and j denotes the class

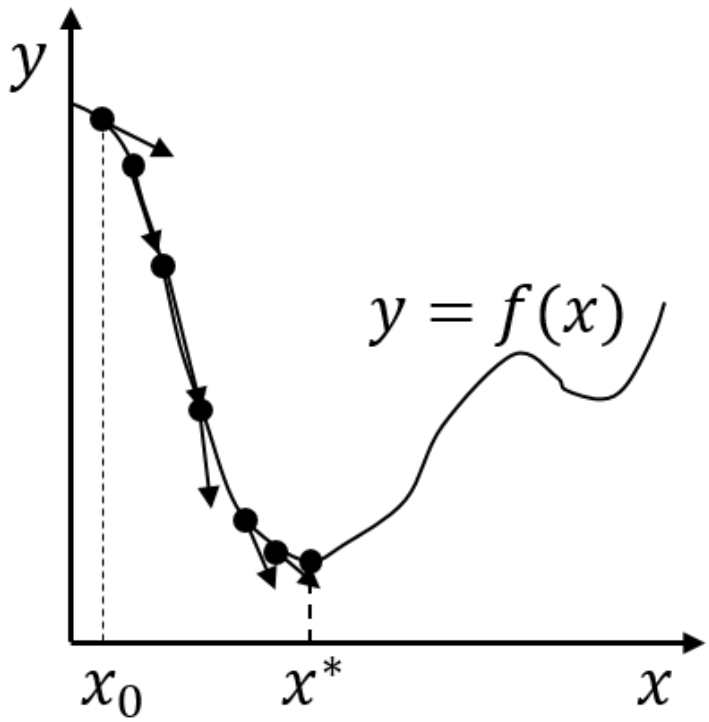
$$L_i = -\sum_j z_{ij} \ln p_{ij} \tag{13}$$

Stochastic gradient descent (1)

- Gradient descent: a generalization of the Newton's method
- Example: find the minimal value of a mono-variable function $y = f(x)$

Algorithm: gradient descent

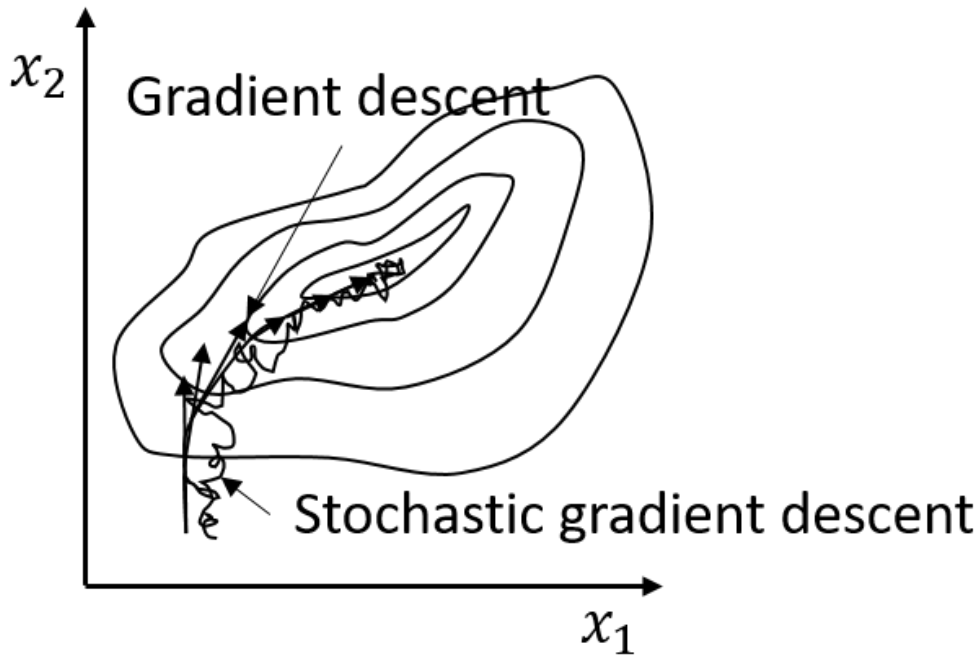
1. Initialize x_0
2. $x_{n+1} = x_n - \alpha f'(x_n)$
3. Stop if $f'(x_n)$ small enough
4. $x^* := x_n$ is the solution



- Pros: the most general method for optimization
- Cons: local extrema, low efficiency, ...

Stochastic gradient descent (2)

- Multi-variable function: in every *epoch*, update all the variables of the function defined by all the training examples through gradient descent (what if the training set is very large?)
- Stochastic gradient descent, incremental gradient descent, or on-line gradient descent: update the gradient sample-by-sample, or batch-by-batch, or by randomly shuffled examples



More techniques on stochastic gradient descent

- Batch normalization: calibrate the offsets and the scales in the feature sets
- Dropout: just update parts of the parameters randomly in every epoch

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

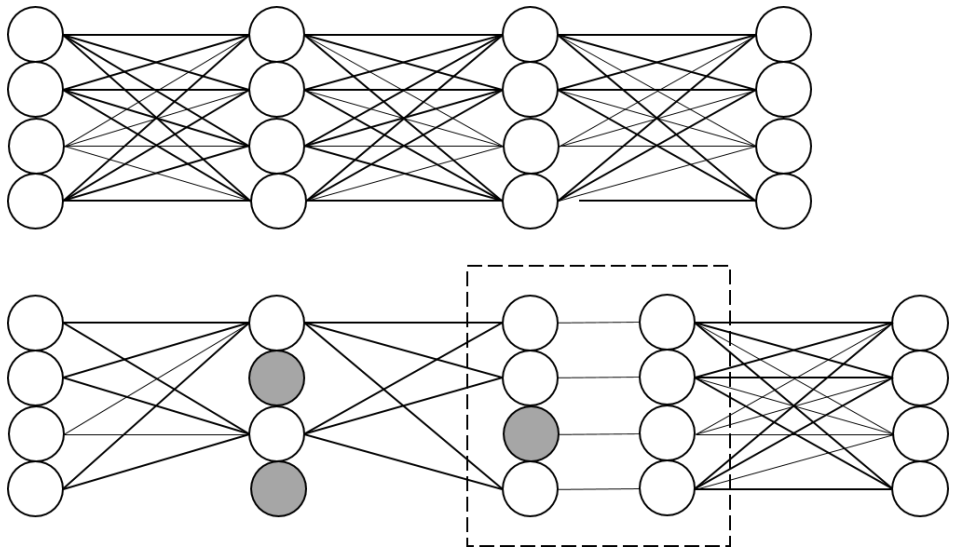
$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.



Sergey Ioffe and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167 (2015).

Backpropagation (1)

- Consider a K-layer DNN, $1 \leq k \leq K$, and $\mathbf{a}^k = \{a_i^k\}_{i=1}^{N^k}$ the kth-layer output, $\mathbf{W} := \{W_{ij}\}_{i=1,j=1}^{N^k,N^{k-1}}$, $\mathbf{b} := \{b_i\}_{i=1}^{N^k}$

$$\mathbf{a}^1 = \mathbf{x} \tag{14}$$

$$\mathbf{z}^k = \mathbf{W}^k \mathbf{a}^{k-)} + \mathbf{b}^k \tag{15}$$

$$\mathbf{a}^k = \sigma(\mathbf{z}^k) \tag{16}$$

$$\mathcal{L}(\mathbf{a}^K, \mathbf{y}) = -\mathbf{y} \ln \mathbf{a}^K - (1 - \mathbf{y}) \ln(1 - \mathbf{a}^K) \tag{17}$$

- Need to compute $\partial \mathcal{L} / \partial \mathbf{W}_{ij}^k$ and $\partial \mathcal{L} / \partial \mathbf{b}_i^k$ for every k
- Variables in one layer are independent from those in another layer
- Derivation using the *chain rule*
- Read the following website for more details:
  `http://neuralnetworksanddeeplearning.com/chap2.html`

An equation for the error in the output layer

Backpropagation (2)

For backpropagation, we have

$$\delta_i^k := \frac{\partial \mathcal{L}}{\partial z_i^k} \tag{18}$$

$$\delta^K = \nabla_a \mathcal{L} \odot \frac{\partial \sigma(z^K)}{\partial z^K} \tag{19}$$

$$\delta^k = \left((\omega^{k+1})^T \delta^{k+1} \right) \odot \frac{\partial \sigma(z^k)}{\partial z^k} \tag{20}$$

Then we have

$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}_i^k} = \delta_i^k \tag{21}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_{ij}^k} = a_i^{k-1} \delta_i^k \tag{22}$$

Unsupervised learning

- k-means
- Dimension reduction: PCA and tSNE
- Autoencoder

k-means clustering

may be applied to dictionary learning

- Given a set of observations $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^d$
- k-means clustering: partition the n observations into $k(\leq n)$ sets $\mathbf{S} = \{S_1, S_2, \cdots, S_k\}$, μ_i be the mean point of S_i, to minimize the within-cluster sum of squares

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mu_i\|_2^2 \tag{23}$$

k-means algorithm: initialization, assignment, update

1. Initialize k mean points $\{\mu_1, \mu_2, \cdots, \mu_k\}$
2. Assign every $\mathbf{x}_i$ to its nearest mean point and construct S_i
3. For every constructed S_i update the mean point μ_i

See Wikipedia for a detailed demonstration!
`https://en.wikipedia.org/wiki/K-means_clustering`

Principal component analysis (PCA)

An operational definition of PCA

- Given a set of observations $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^p$
- Let $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times p}$
- The covariance matrix $\mathbf{C} = \mathbf{X}^T\mathbf{X}/(n-1) \in \mathbb{R}^{p \times p}$
- The eigenvalues of $\mathbf{C}$ are called principal axes or principal directions
- Projections of the data on the principal axes are called principal components or PC scores
- $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$, $\mathbf{V} := [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n]$ are principal axes, $\mathbf{XV}$ principal components
- $\mathbf{v}_i$ with larger eigenvalues are more important

tSNE

- t-Distributed Stochastic Neighbor Embedding
- Solve the underfitting issue of PCA
- Better for visualization!
- Example: `https://www.youtube.com/watch?v=C4h-iQdL3M4`
- See sklearn.manifold.TSNE

Autoencoder

- An *encoder* plus a *decoder* (usually symmetric)
- Train the input **x** to fit itself
- Bottleneck feature as a *latent representation*

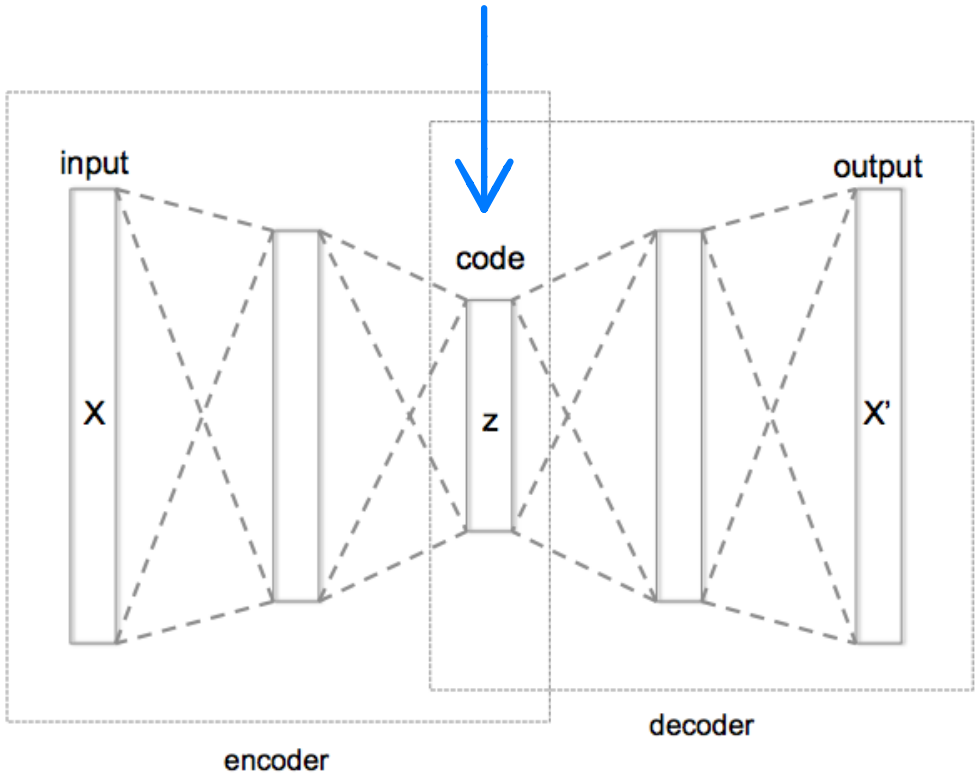
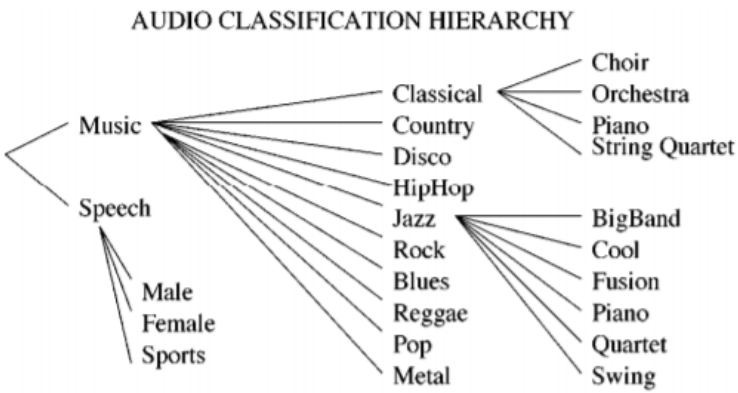


Figure from Wikipedia

Musical genre classification

- Feature extraction + classification
- Timbral texture features: spectral centroid, spectral roll-off, ...
- Rhythmic content features: beat histogram
- Pitch contour features: pitch salience function
- Classifier: Gaussian mixture models (GMM), kNN, ...
- GTZAN dataset: 1000 30-sec music clips, 10 genres (classical, country, disco, hip-hop, jazz, rock, blues, reggae, pop, metal)



Confusion Matrix

	cl	co	di	hi	ja	ro	bl	re	po	me
cl	**69**	0	0	0	1	0	0	0	0	0
co	0	**53**	2	0	5	8	6	4	2	0
di	0	8	**52**	11	0	13	14	5	9	6
hi	0	3	18	**64**	1	6	3	26	7	6
ja	26	4	0	0	**75**	8	7	1	2	1
ro	5	13	4	1	9	**40**	14	1	7	33
bl	0	7	0	1	3	4	**43**	1	0	0
re	0	9	10	18	2	12	11	**59**	7	1
po	0	2	14	5	3	5	0	3	**66**	0
me	0	1	0	1	0	4	2	0	0	**53**

George Tzanetakis and Perry Cook, "Musical genre classification of audio signals," IEEE Transactions on speech and audio processing 10.5 (2002): 293-302.

Instrument classification

Many about musical instruments

- Instrument family / instrument type
- Instrument playing technique
- Predominant instrument recognition in polyphonic music
- Datasets: Good-sounds 樂器 & 演奏 Quality (http://mtg.upf.edu/download/datasets/good-sounds), IOWA (http://theremin.music.uiowa.edu/MIS.html), etc.

Table 2: Classification results

	Hierarchy 1	Hierarchy 2	No hierarchy
Pizzicato / sustained	99.0%	99.0%	99.0%
Instrument families	93.0%	94.0%	94.7%
Individual instruments	74.9%	75.8%	80.6%

Antti Eronen and Anssi Klapuri, "Musical instrument recognition using cepstral coefficients and temporal features," IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000.
https://labrosa.ee.columbia.edu/~dpwe/e6820/papers/EroK00-inst.pdf

Musical emotion recognition 難度較高的 topic

人聽到的感受 不一定等於 寫曲人的想法

- **Categorical psychometrics**: happy, sad, ...
- **Scalar/dimensional psychometrics**: Valence-Arousal (V-A) plane (Russell and Thayer, 1980)
- Emotions existing on a plane along independent axes of arousal (intensity), ranging high-to-low, and valence (an appraisal of polarity), ranging positive-to-negative

Yi-Hsuan Yang *et al.* "A regression approach to music emotion recognition," IEEE Transactions on audio, speech, and language processing 16.2 (2008): 448-457.

Musical emotion recognition: datasets

Multi-class problem for categorical psychometrics, and regression problem for scalar psychometrics

- AMG1608 Dataset: 1608 30-second music clips with dimensional annotation (`http://mpac.ee.ntu.edu.tw/dataset/AMG1608/`)
- CISUC multi-modal dataset: 904 Western pop music, with categorical annotation (`http://mir.dei.uc.pt/downloads.html`)
- and more

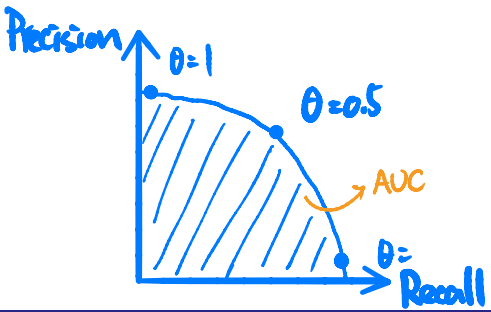
Figure of merit (1)

- True Positive (TP): True is identified True (correct identification)
- True Negative (TN): False is identified False (correct identification)
- False Positive (FP): True is identified False (wrong identification)
- False Negative (FN): False is identified True (wrong identification)
- Precision (P) = TP/(TP+FP)
- Recall (R) = TP/(TP+FN)
- F-score (F) = 2PR/(P+R) 調和平均

Figure of merit (2)

- For a binary classifier with output z between 0 and 1, we always need to choose a threshold $\theta \in [0, 1]$ to determine the true or the false
- Performance depends on θ

  - High θ: high precision, low recall
  - Low θ: low precision, high recall
  - Receiver operating characteristic (ROC)
  - Area under the ROC curve (AUC)



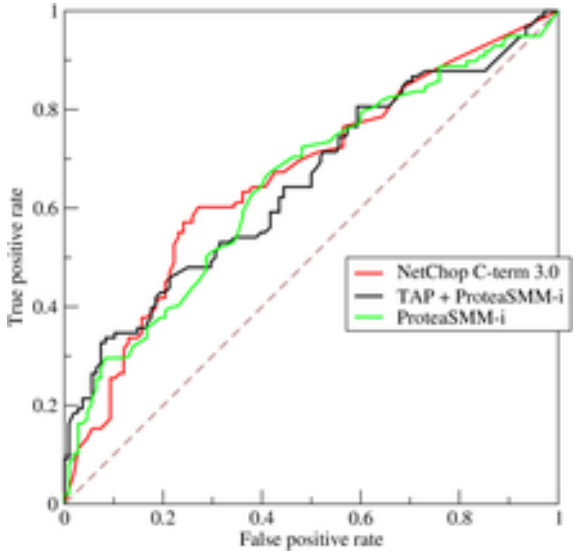


Figure from Wikipedia

Auto-tagging

- Tags: a number of high-level concepts from potentially very diverse music facets
- Emotion, Musical instruments, Genre, Usage, etc.
- Multi-label classification problems
- Datasets:
  - The Computer Audition Lab 500 (CAL500) dataset (`http://cosmal.ucsd.edu/cal/projects/AnnRet/`)
  - The CAL10k dataset (`http://calab1.ucsd.edu/~datasets/`)
  - The Magnatagatune dataset (`http://tagatune.org/Magnatagatune.html`)
- An example of professionally annotated tags:

Category	Tags
Genre	bossanova, country rock, hymn, orchestral pop, slide blues
Mood	alarm, awards, catchy, danger, glamour, military, scary, smooth, trance

Markus Schedl *et al.* "A professionally annotated and enriched multimodal data set on popular music," Proceedings of the 4th ACM Multimedia Systems Conference, 2013.

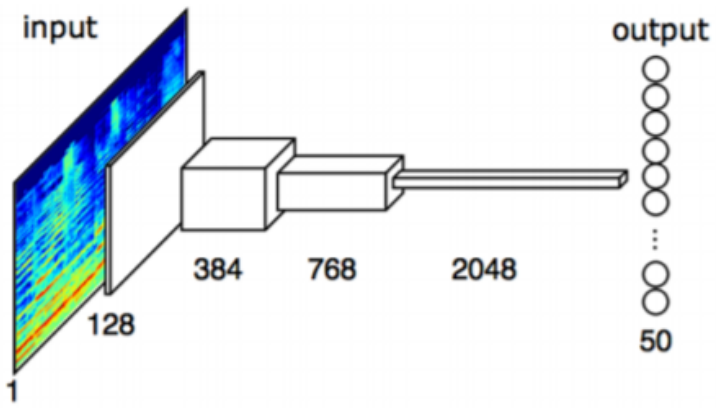
An example on feature and tag

Acoustic tags most improved by BDS classifier		
Tag	Improvement (AUC)	First 5 Features Selected
"triple note feel"	0.1926	time signature, time signature, variance of bar length, ratio of beat length to bar length, tatums per second
"a twelve-eight time signature"	0.1816	key, time signature, variance of bar length, weighted ratio of beat length to bar length, song loudness
"minimalist arrangements"	0.1732	number of sections, song loudness, weighted ratio of tatum length to beat length, mode, end of fade in
"a mid-tempo shuffle feel"	0.0828	song loudness, time signature, start of fade out, weighted ratio of beat length to bar length, variance of tatum length
"use of modal harmony"	0.0782	song loudness, number of sections, variance of section length, number of sections, key confidence

Derek Tingle, Youngmoo E. Kim, and Douglas Turnbull, "Exploring automatic music annotation with acoustically-objective tags." Proceedings of the international conference on multimedia information retrieval (ICMR), 2010.

Auto-tagging using deep learning

- Deep fully convolutional neural networks (FCN)
- System design processes:
  - Data representation – time-frequency representation
  - Convolution – kernel sizes and axes
  - Pooling – sizes and axes



少見的tag仍有進步空間

Methods	AUC
The proposed system, FCN-4	.894
2015, Bag of features and RBM [18]	.888
2014, 1D convolutions [6]	.882
2014, Transferred learning [28]	.88
2012, Multi-scale approach [5]	.898
2011, Pooling MFCC [8]	.861

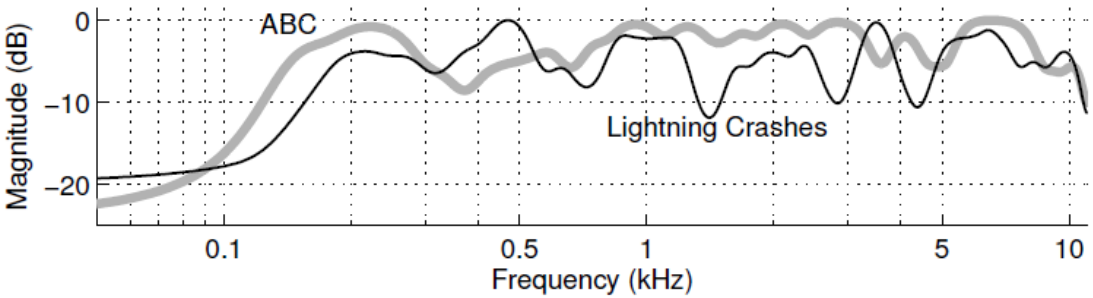
Keunwoo Choi, George Fazekas, and Mark Sandler, "Automatic tagging using deep convolutional neural networks," in ISMIR 2016.

Keunwoo Choi *et al.* "A tutorial on deep learning for music information retrieval," arXiv preprint arXiv:1709.04396, 2017.

Some critics

MIR community 較不愛 DNN, 因為 musician 想調控想理解 knowledge

- Does a genre classification method really learns what a genre means? → 所有 data driven 的方法都有這問題
- The "Clever Hans" problem in machine learning (Strum, 2012)
- In other words, (all) genre classification systems can easily be fooled
- Example: change a happy song to a sad one using an equalizer



Bob L. Sturm, "Evaluating music emotion recognition: Lessons from music genre recognition?" IEEE Int. Conf. on Multimedia and Expo Workshops (ICMEW), 2013.

Bob L. Sturm, "The future of scientific research in music information retrieval."

Music + EQ can fool the model, 過不了音樂製作人的關, 人們無法合理調理

Challenges

Challenges

- Global and local properties
- Co-occurrence and separability
- Weakly-supervised learning

Notes

- Domain knowledge is important
- Signal processing is important → 使機器更能真正理解
- Evaluation is important

Resources

- Light-weighted machine learning in Python: `scikit-learn`
- Deep learning courses: Hung-yi Lee's YouTube lectures