# Deep Learning Techniques for Music Generation (3)

Jean-Pierre Briot

Jean-Pierre.Briot@lip6.fr

Laboratoire d'Informatique de Paris 6 (LIP6)

Sorbonne Université – CNRS

Programa de Pós-Graduação em Informática (PPGI)

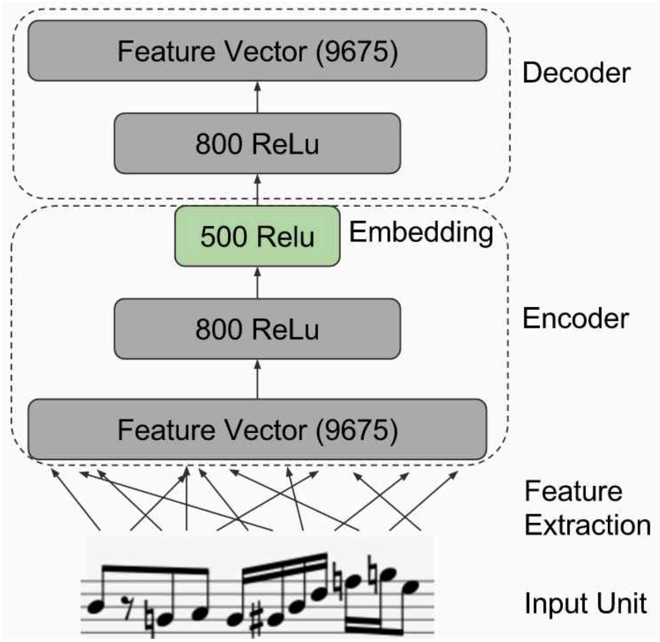
UNIRIO

# Feedforward

# Deep Learning Techniques for Computer Music

- Motivation:
  - Multiple massive corpora available
  - Mature architectures and efficient implementations

- Ex: Features extraction
  - Ex: for Music Information Retrieval
  - Past: Manual rich representations
  - Now: Automatic features extraction
  - (using ex: Stacked Autoencoders)




[Bretan et al., 2016]

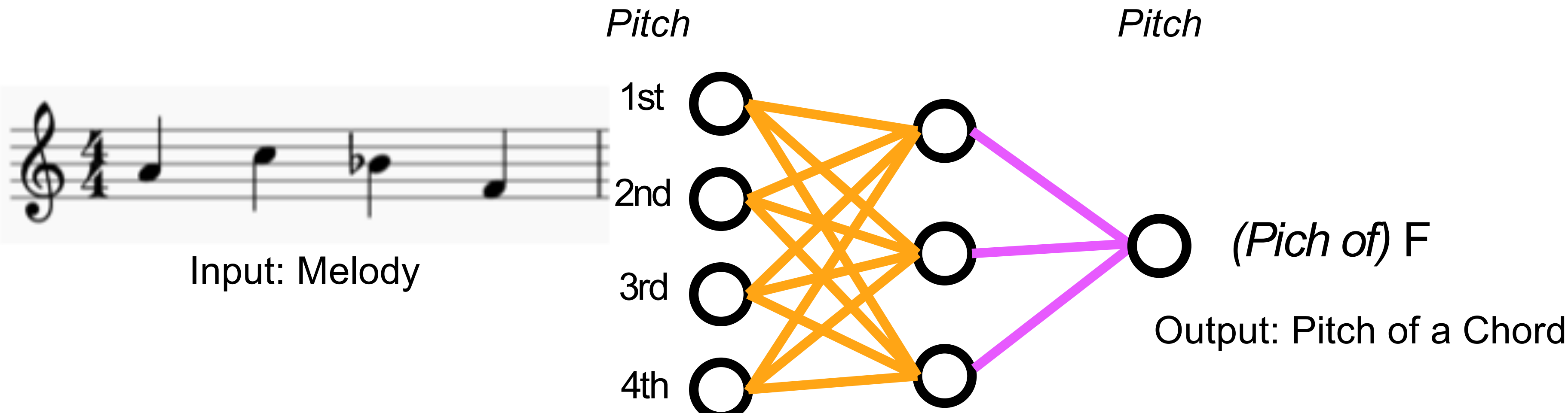
- Other models/techniques
  - Human-specified:
    - » Grammars, Rules...
  - Learnt:
    - » Automata, Markov models, Graphical models, Entropy models…

# Deep Learning for Music Generation – Technical Challenges

1. ***Ex Nihilo* Generation**
   » vs Accompaniment (Need for Input)
2. **Length Variability**
   » vs Fixed Length
3. **Content Variability**
   » vs Determinism
4. **Control**
   » ex: Tonality conformance, Maximum number of repeated notes…
5. **Structure**
6. **Originality**
   » vs Conformance
7. **Incrementality**
   » vs Single-step or Iterative Generation
8. **Interactivity**
   » vs (Autistic) Automation
9. **Explainability**

# Direct Use – Feedforward – Ex 1

- Feedforward Architecture
- Prediction (or Classification) Task
- Ex1: Predicting a chord associated to a melody segment
  – scale/mode -> tonality

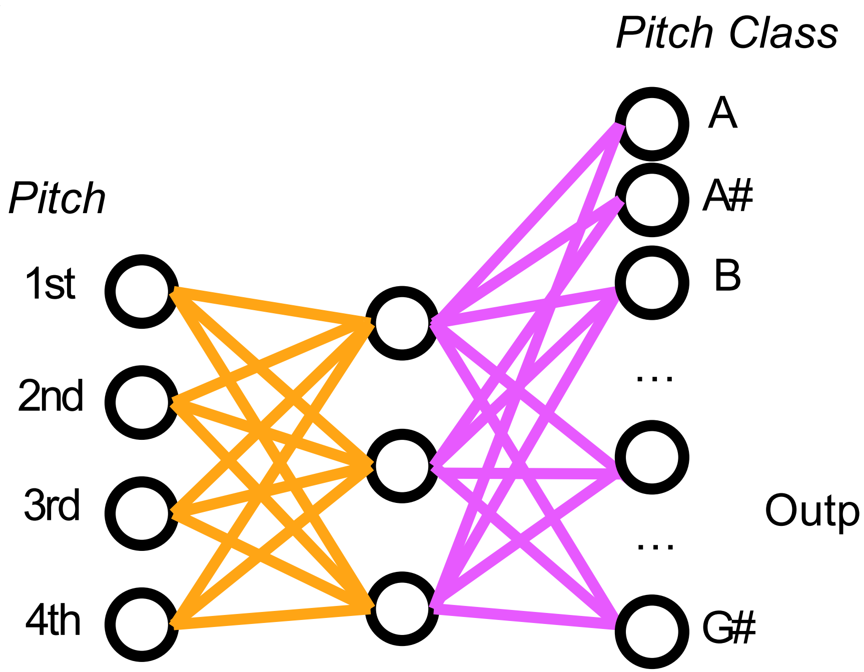



  – Training on a corpus/dataset <melody, chord>
  – Production (Prediction)

# Direct Use – Feedforward – Ex 1

- Feedforward Architecture
- Classification Task
- Ex1: Predicting a chord associated to a melody segment
  - scale/mode -> tonality

Input: Melody




F

Output: Chord (Pitch Class)

  - Training on a corpus/dataset <melody, chord>
  - Production (Classification)

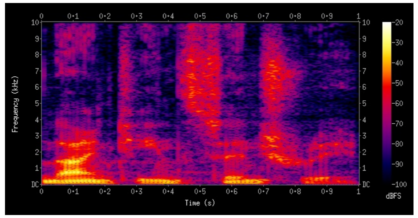
# Code

- Python (3)
- Keras
- Theano
- Music21

# Representation

- Audio
  - Waveform

  - Spectrogram (Fourier Transform)

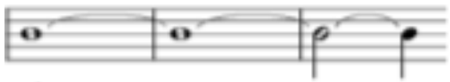


  - Other (ex: MFCC)

- Symbolic
  - Note

  - Rest

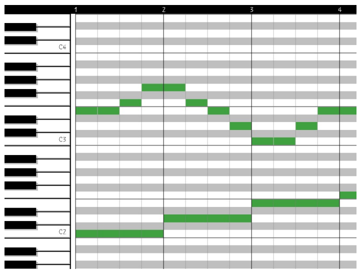
  - Note hold



  - Duration
  - Chord

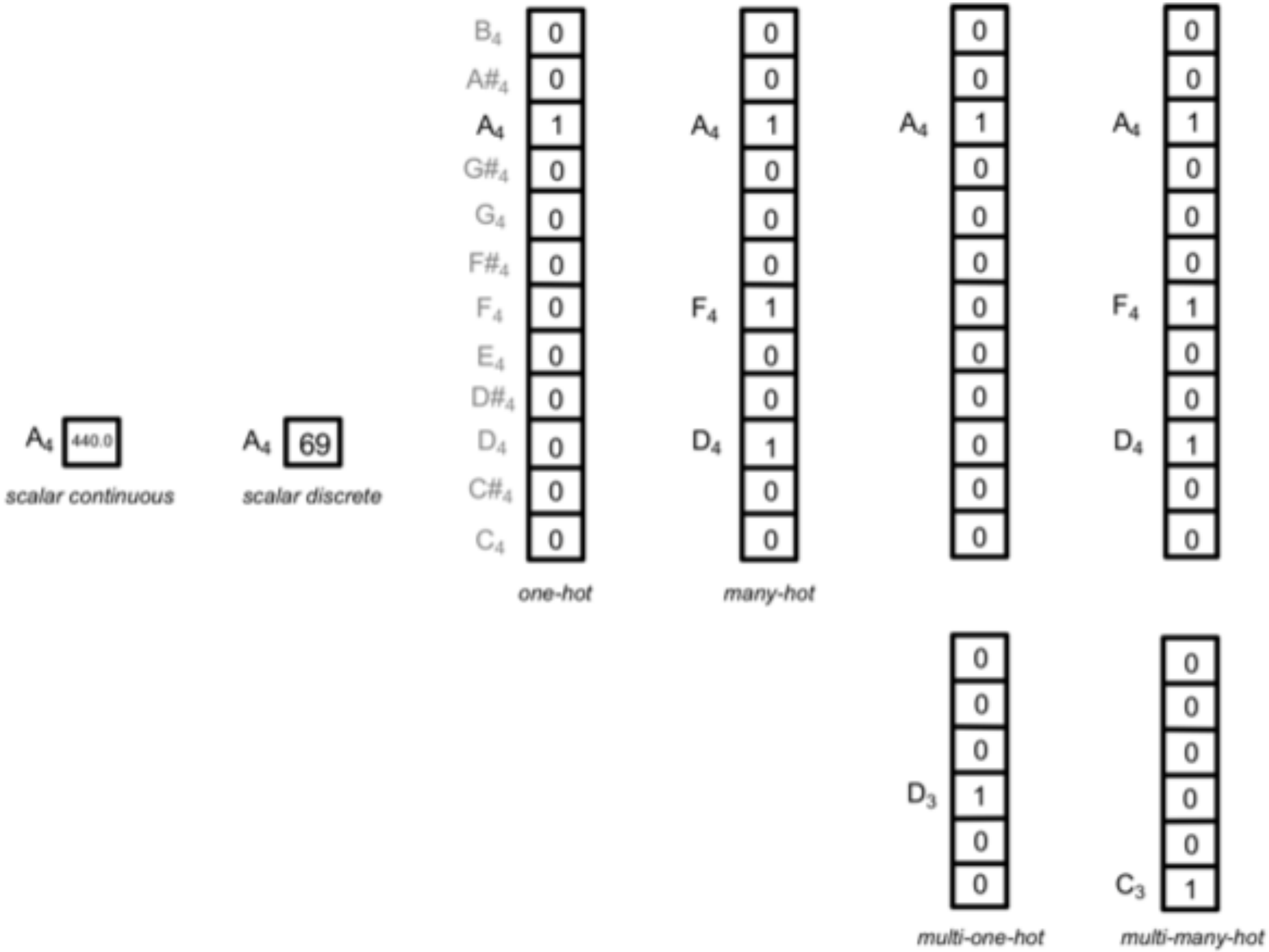
  - Rhythm

  - Piano Roll



# Encoding of Features (ex : Note Pitch)

- Value
  - *Analogic*
- One-Hot
  - *Digital*
- Embedding
  - *Constructed*

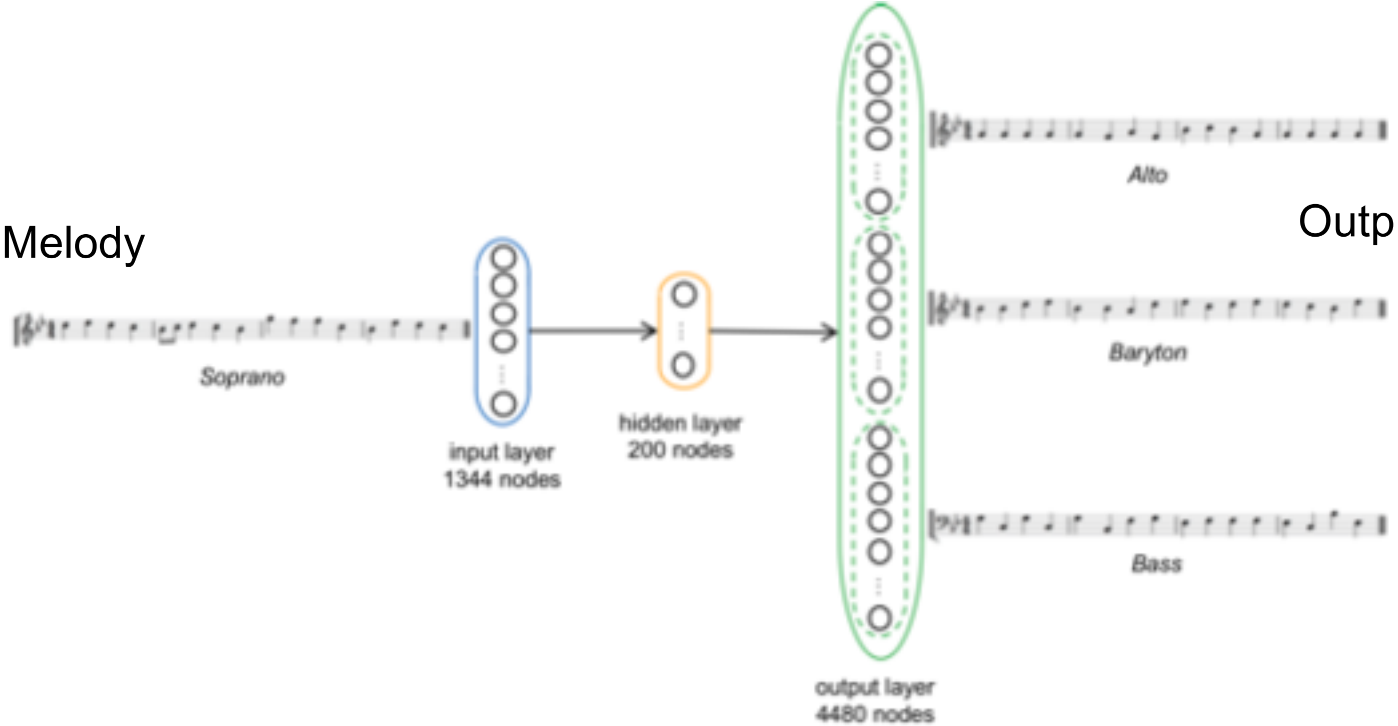


# Direct Use – Feedforward – Ex 2: MiniBach

- Feedforward Architecture
- Prediction Task
- Ex2: Counterpoint (Chorale) generation
- MiniBach [Hadjeres & Briot, 2017]
- Training on the set of (389) J.S. Bach Chorales (Choral Gesang)

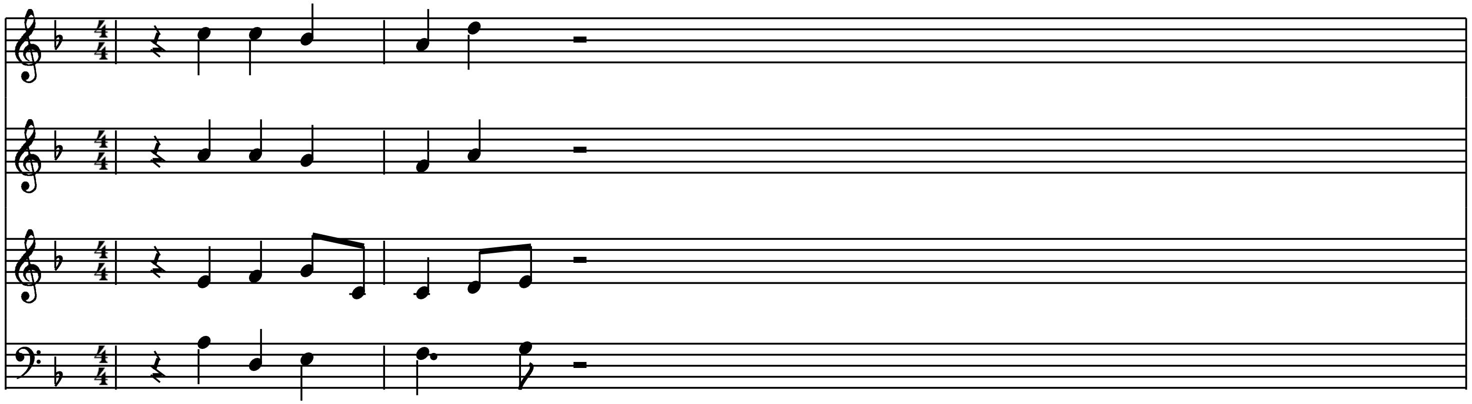

Input: Melody

Output: 3 Melodies

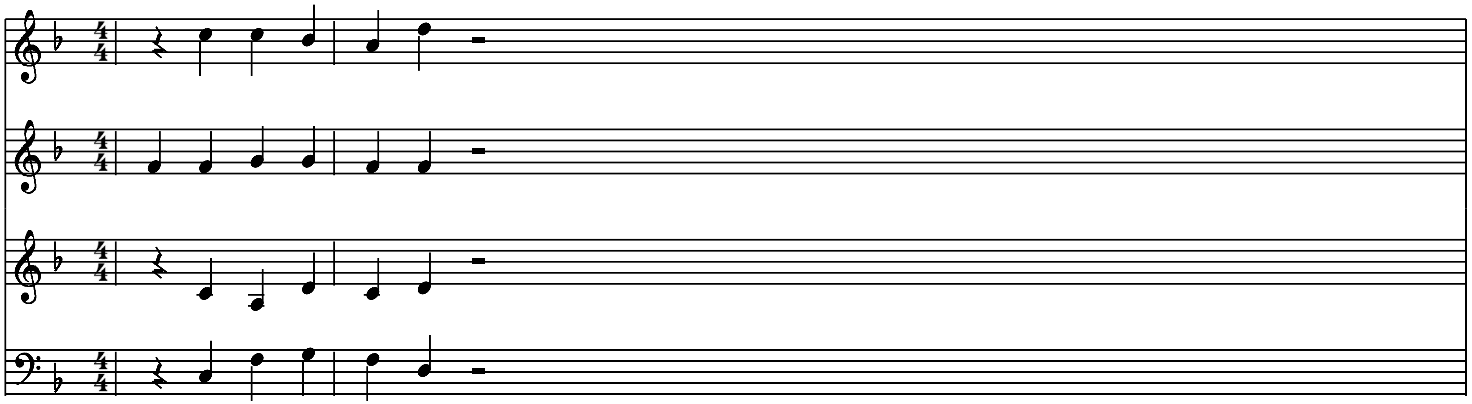


# MiniBach Code & Examples

minibach-original-chorale



minibach-predicted-chorale



# Output Activation and Cost/Loss Functions

| *Task* | *Output type* | *Encoding of the target* | *Output activation function* | *Cost (loss)* | *Application* |
|---|---|---|---|---|---|
| Regression | Numerical value | $\mathbb{R}$ or $\mathbb{I}$ | Identity (Linear) | Quadratic cost | |
| Classification | Binary value | 0 or 1 | Sigmoid | Binary cross-entropy | |
| Classification | Multiclass single label | One-hot | Softmax | Categorical cross-entropy | Monophony |
| Classification | Multiclass multilabel | Many-hot | Sigmoid | Binary cross-entropy | Polyphony |
| Multiple Classification | Multi multiclass single label | Multi-one-hot | Sigmoid | Binary cross-entropy | Multivoice |

Ex multiclass single label: Classification among a set of possible notes for a monophonic melody, with only one single possible note choice (single label)

Ex multiclass multilabel: Classification among a set of possible notes for a single-voice polyphonic melody, therefore with several possible note choices (several labels)

Ex multi multiclass single label: Multiple classification among a set of possible notes for a set of time slices for a monophonic melody, therefore for each time slice with only one single possible note choice

# Output Activation and Cost/Loss Functions

| *Task* | *Output type* | *Encoding of the target* | *Output activation function* | *Cost (loss)* | *Interpretation* |
|---|---|---|---|---|---|
| Regression | Numerical value | $\mathbb{R}$ or $\mathbb{I}$ | Identity (Linear) | Quadratic cost | none |
| Classification | Binary value | 0 or 1 | Sigmoid | Binary cross-entropy | none |
| Classification | Multiclass single label | One-hot | Softmax | Categorical cross-entropy | argmax or sampling |
| Classification | Multiclass multilabel | Many-hot | Sigmoid | Binary cross-entropy | argsort and > threshold & max-notes |
| Multiple Classification | Multi multiclass single label | Multi-one-hot | Sigmoid | Binary cross-entropy | *p* argmax or sampling |

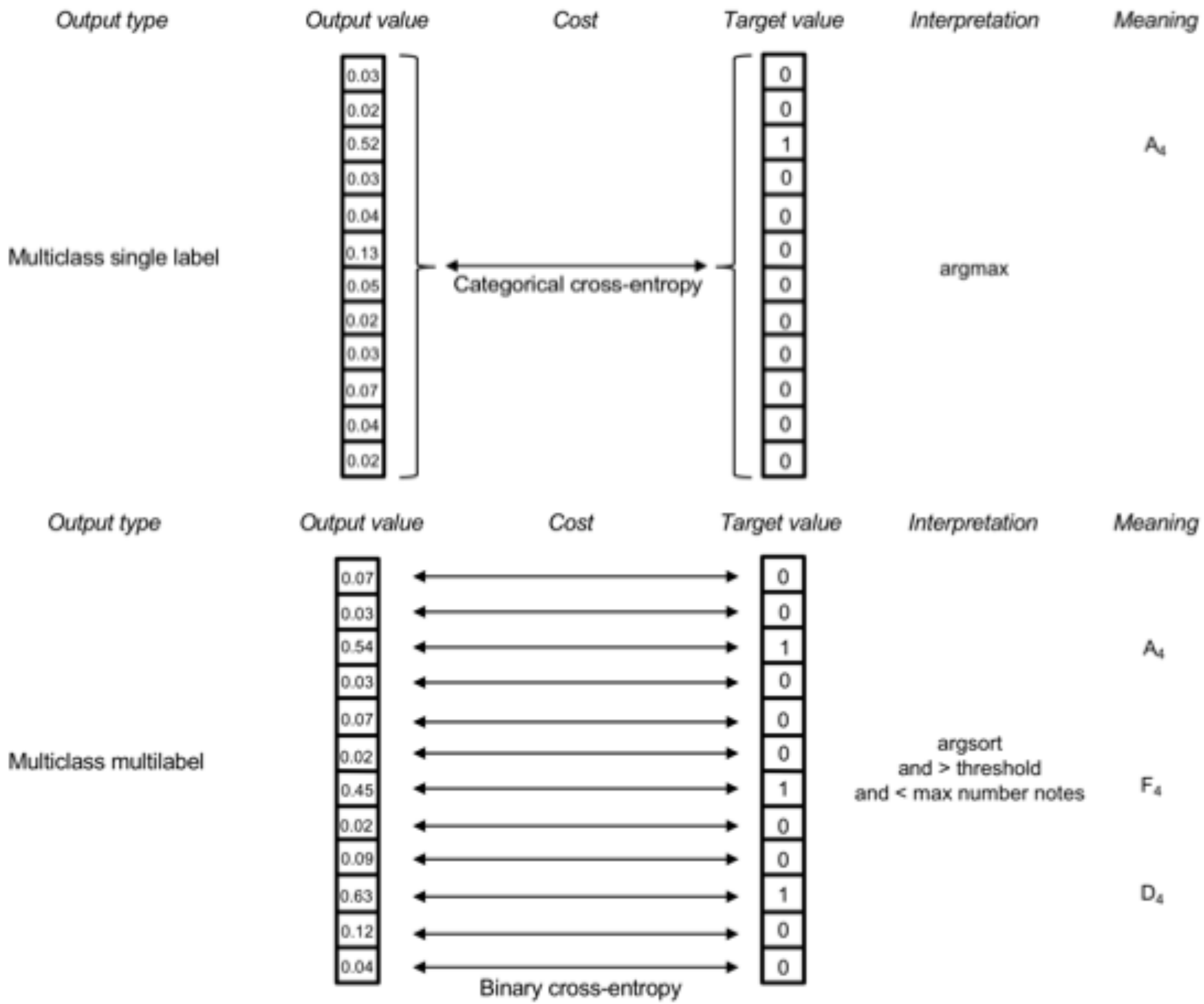
*Other cost functions:*

Mean absolute error, Kullback-Leibler (KL) divergence…

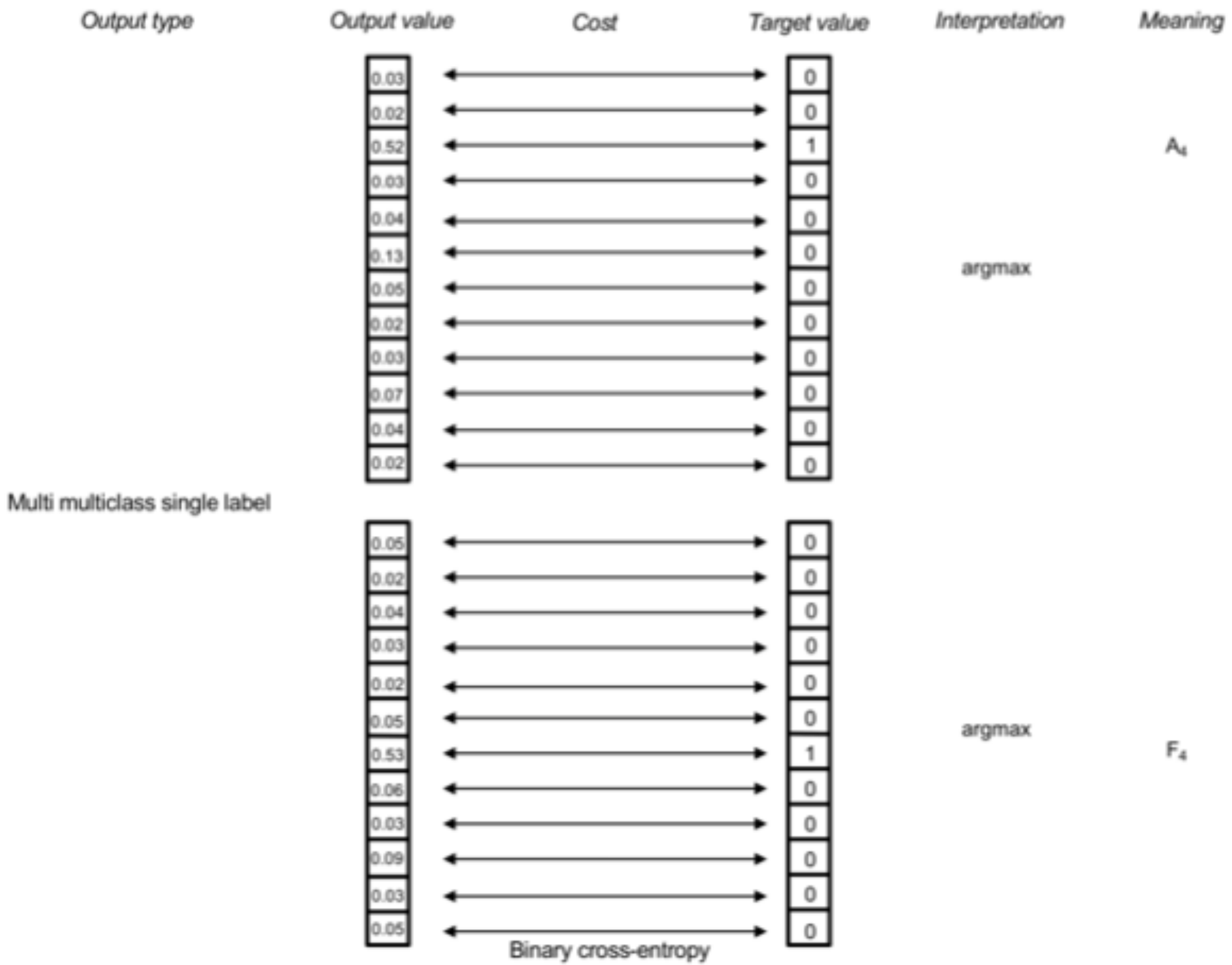
# Output Activation and Cost/Loss Functions

| *Output type* | | *Output value* | *Cost* | *Target value* | *Interpretation* | *Meaning* |
|---|---|---|---|---|---|---|
| Numerical | Real | 439.7 | Mean squared error | 440.0 | | $A_4$ |
| | Integer | 69.2 | Mean squared error | 69 | | $A_4$ |
| Binary | | 0.97 | Binary cross-entropy | 1 | > 0.5 | True |

# Output Activation and Cost/Loss Functions



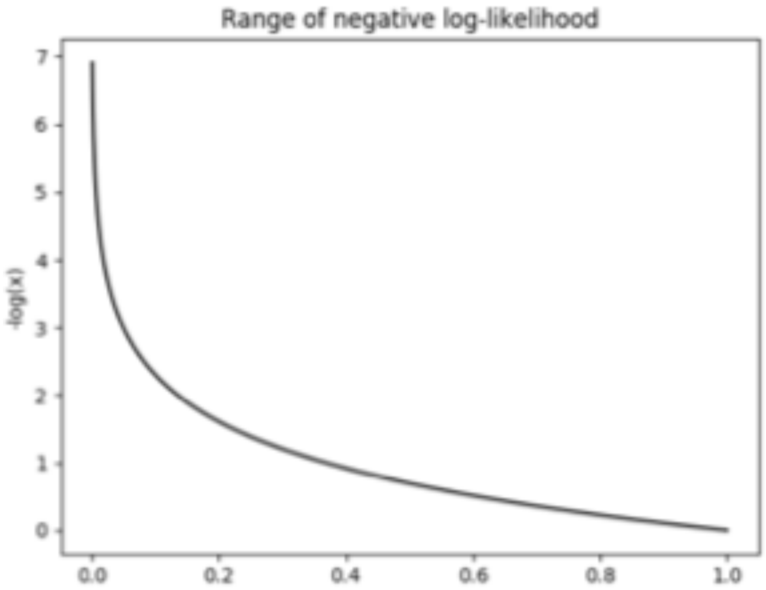
# Output Activation and Cost/Loss Functions



# (Summary of) Principles of Loss Functions

- Probability theory + Information theory
- Probability : $P(x)$
- Self-information : $I(x) = -\log P(x)$
- Entropy : $H(\mathrm{x}) = \mathrm{E}_{\mathrm{x}\sim P}\, I(x) = \mathrm{E}_{\mathrm{x}\sim P} - \log P(x)$
- $\mathrm{E}_{\mathrm{x}\sim P}\, f(\mathrm{x})$ : Expectation : mean value of f(x) when x~$P$

  $= \sum_x P(x)\, f(x)$



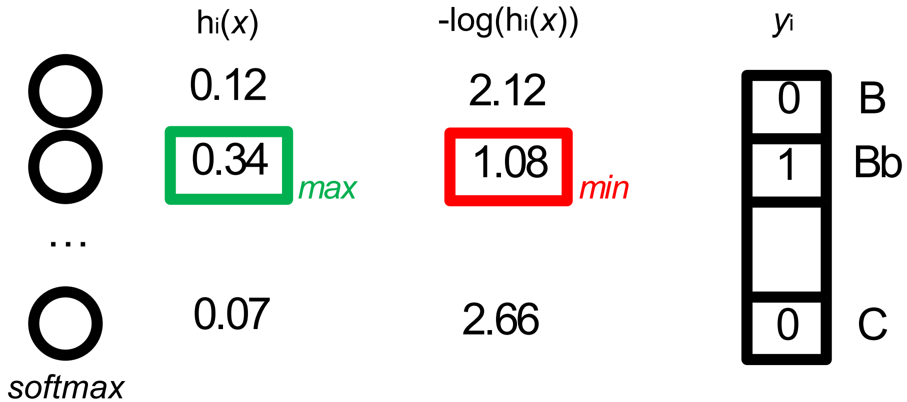

- Cross entropy : $H(P, Q) = -\mathrm{E}_{\mathrm{x}\sim P} \log Q(x)$

  $H(\mathrm{h}(x), y) = -\sum_i \mathrm{h}_i(x) \log y_i$

- Kullback-Leibler Divergence : $D_{KL}(P \,||\, Q) = \mathrm{E}_{\mathrm{x}\sim p}\, [\log (P(x) / Q(x))]$

  $= \mathrm{E}_{\mathrm{x}\sim p}\, [\log P(x) - \log Q(x)]$

- *Maximum likelihood principle*
- *Minimum Negative likelihood*



# Classifier Evaluation Metrics

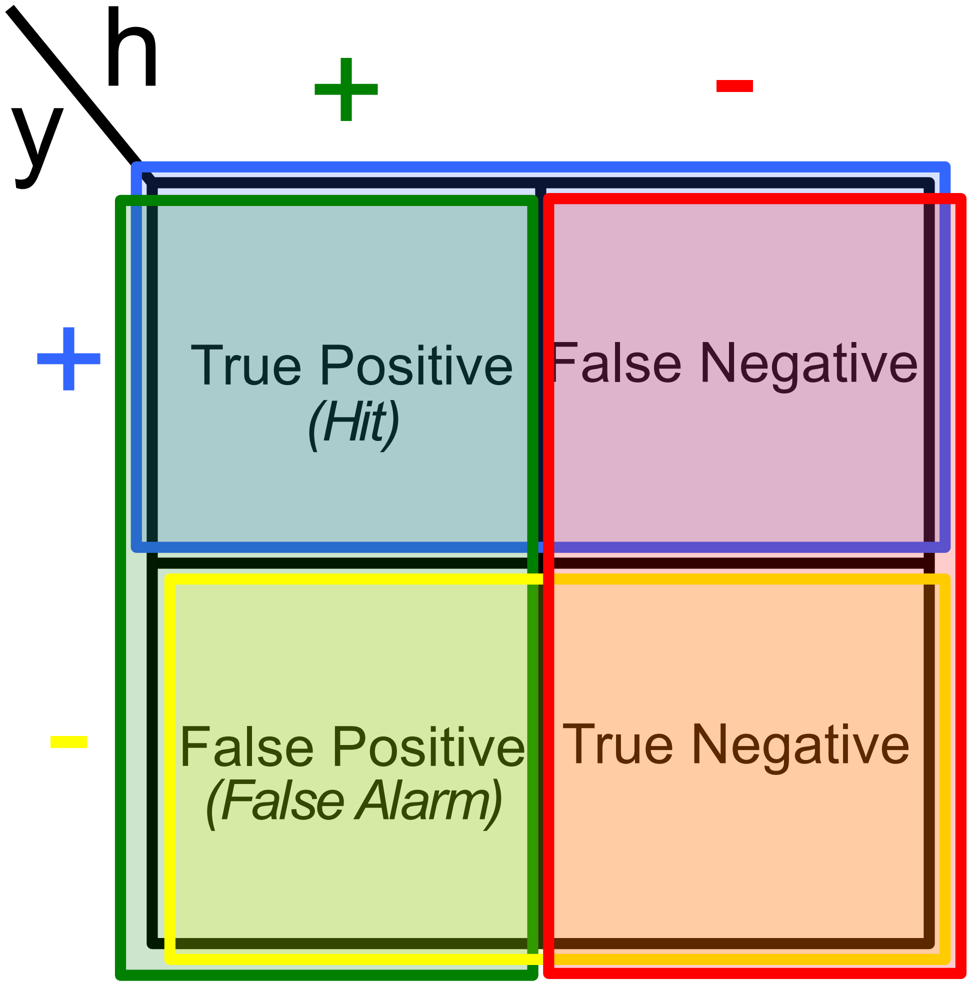
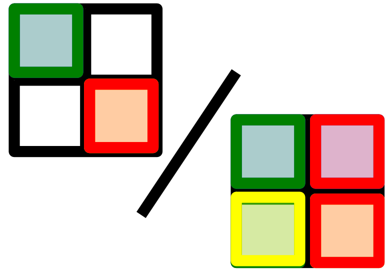
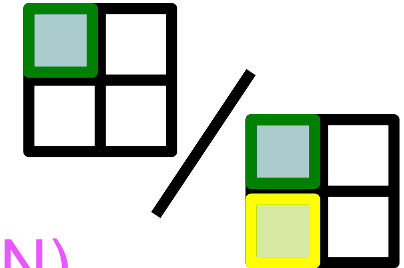



Table of Confusion
(Confusion Matrix)

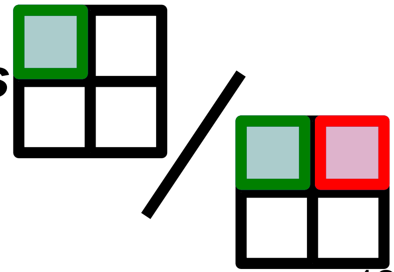
Accuracy = TP + TN / ∀
% correctly classified



Precision = TP / P (TP + FP)
true / classified positive
***How relevant the results***



Recall = TP / P (TP + FN)
true / real positive
***How many relevant results***



# Skewed Classes

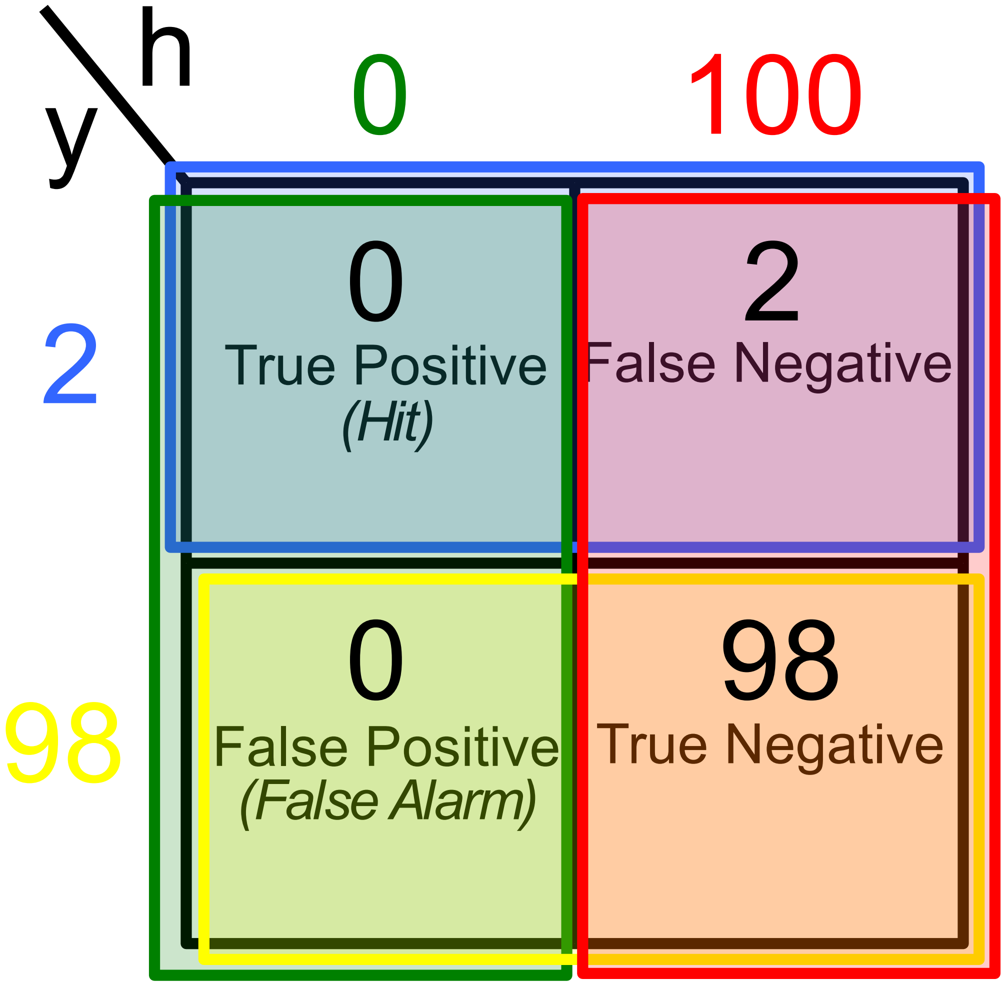
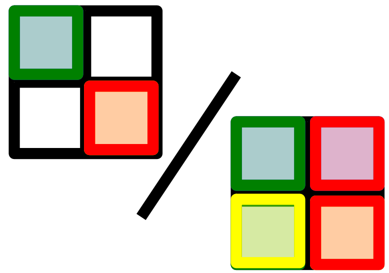
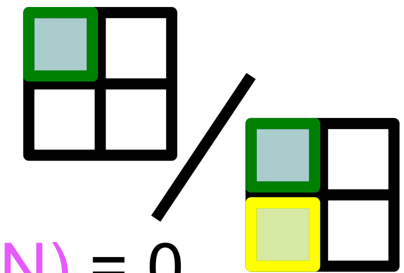



Table of Confusion
(Confusion Matrix)

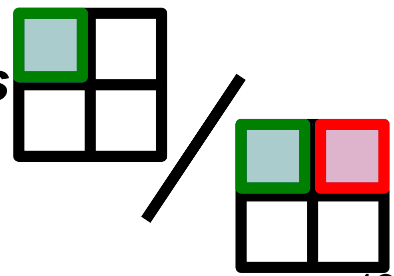
Accuracy = TP + TN / ∀ = 0.98
% correctly classified



Precision = TP / P (TP + FP) = 0
true / classified positive
***How relevant the results***



Recall = TP / P (TP + FN) = 0
true / real positive
***How many relevant results***



Ex: Maligne Cancer Prediction: 2% Rate
Optimist Hypothese: Always (100%) False

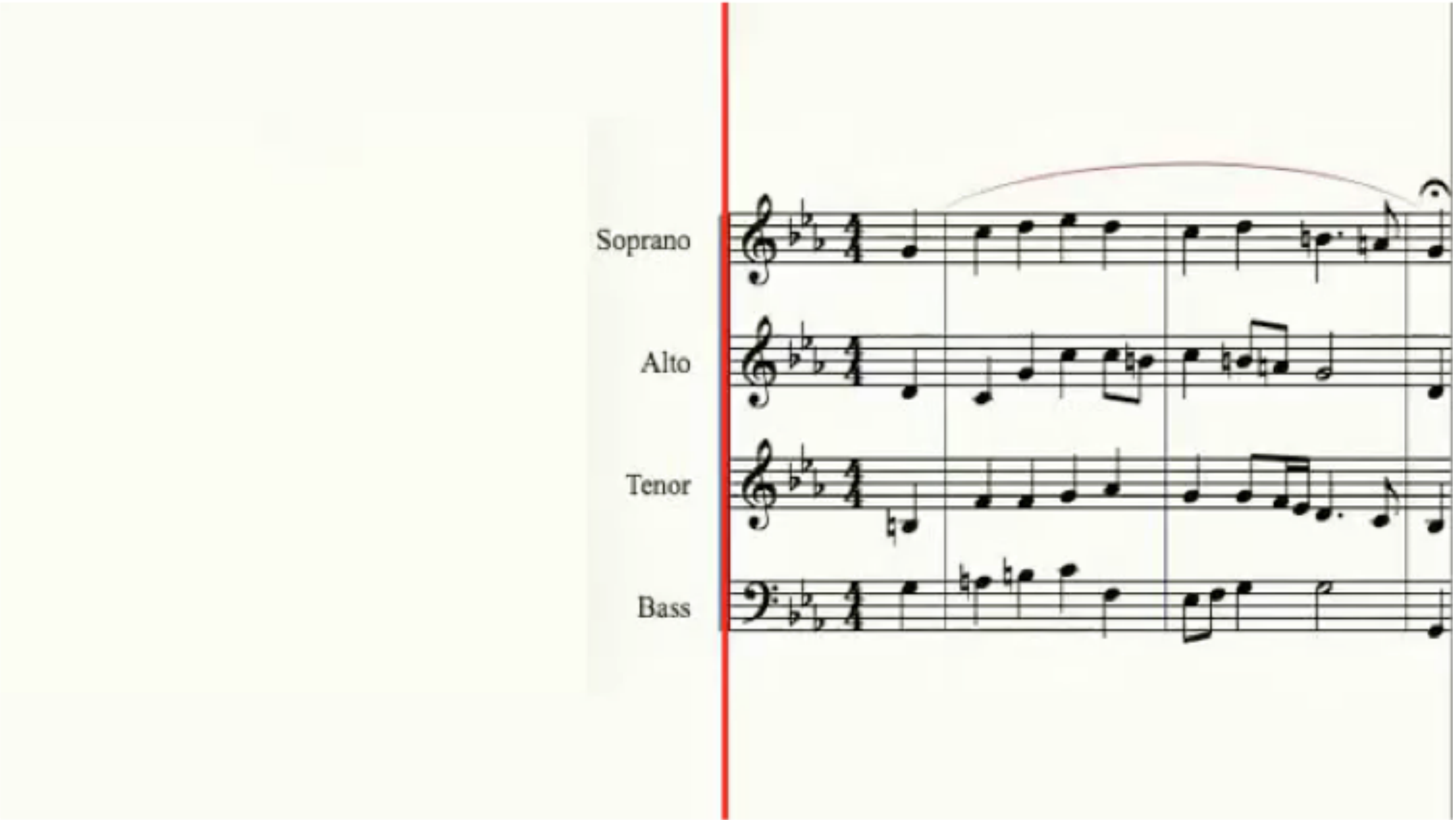
# MiniBach – Ex. of Result

# MiniBach vs DeepBach

MiniBach

DeepBach

# ~~Mini~~DeepBach – Demo




https://www.youtube.com/watch?v=QiBM7-5hA6o

# Reorchestration of Ode of Joy by DeepBach (and other techniques [Flow Machines])

Ode to Joy in several styles

https://www.youtube.com/watch?v=buXqNqBFd6E

# Reorchestration of God Save the Queen by DeepBach

https://www.youtube.com/watch?time_continue=1&v=x-W0ixD9Cpg