Deep learning in practice
a Text-to-Speech scenario

6th Deep Learning Meetup

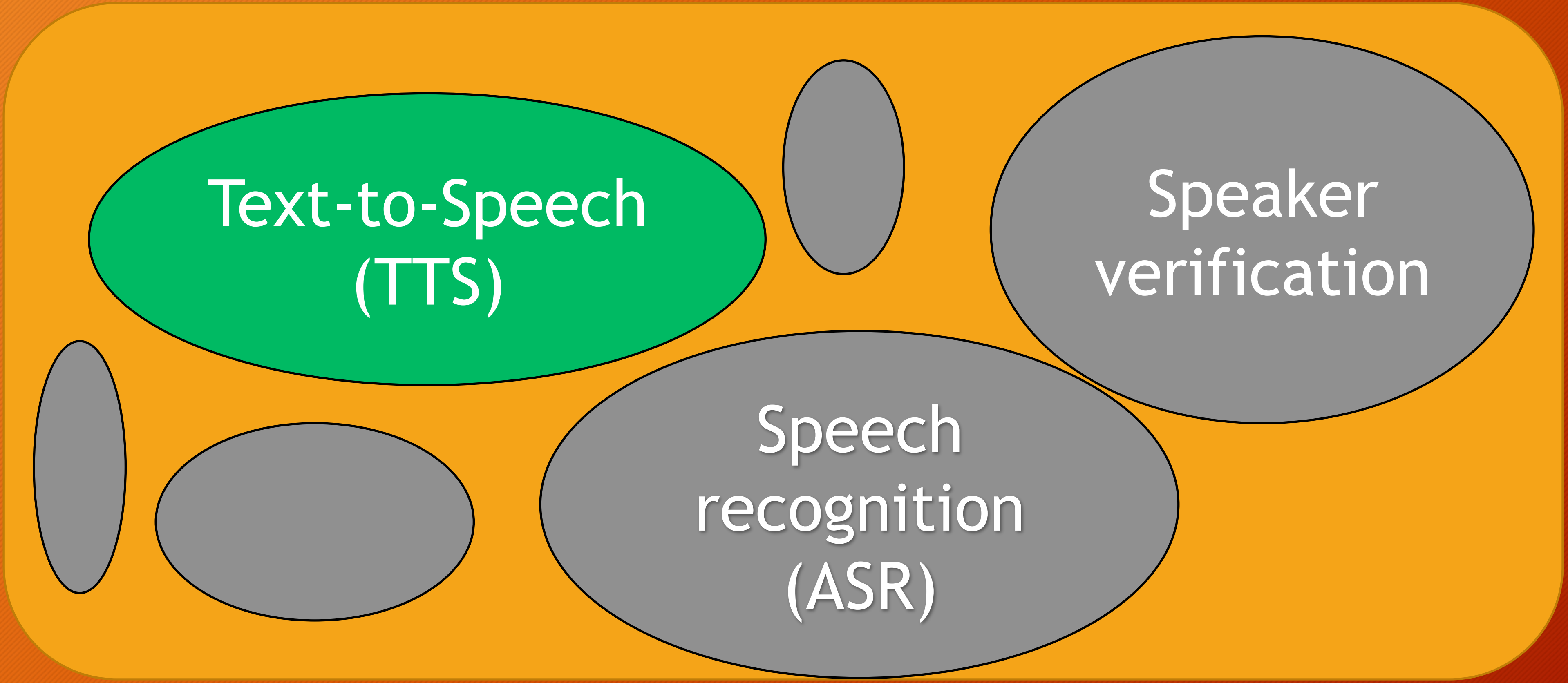
Kornel Kis

Vienna, 12.10.2016.

Main Topics

- Speech technology (fundamentals)
- Deep learning in practice
- Ensemble learning (a bit)

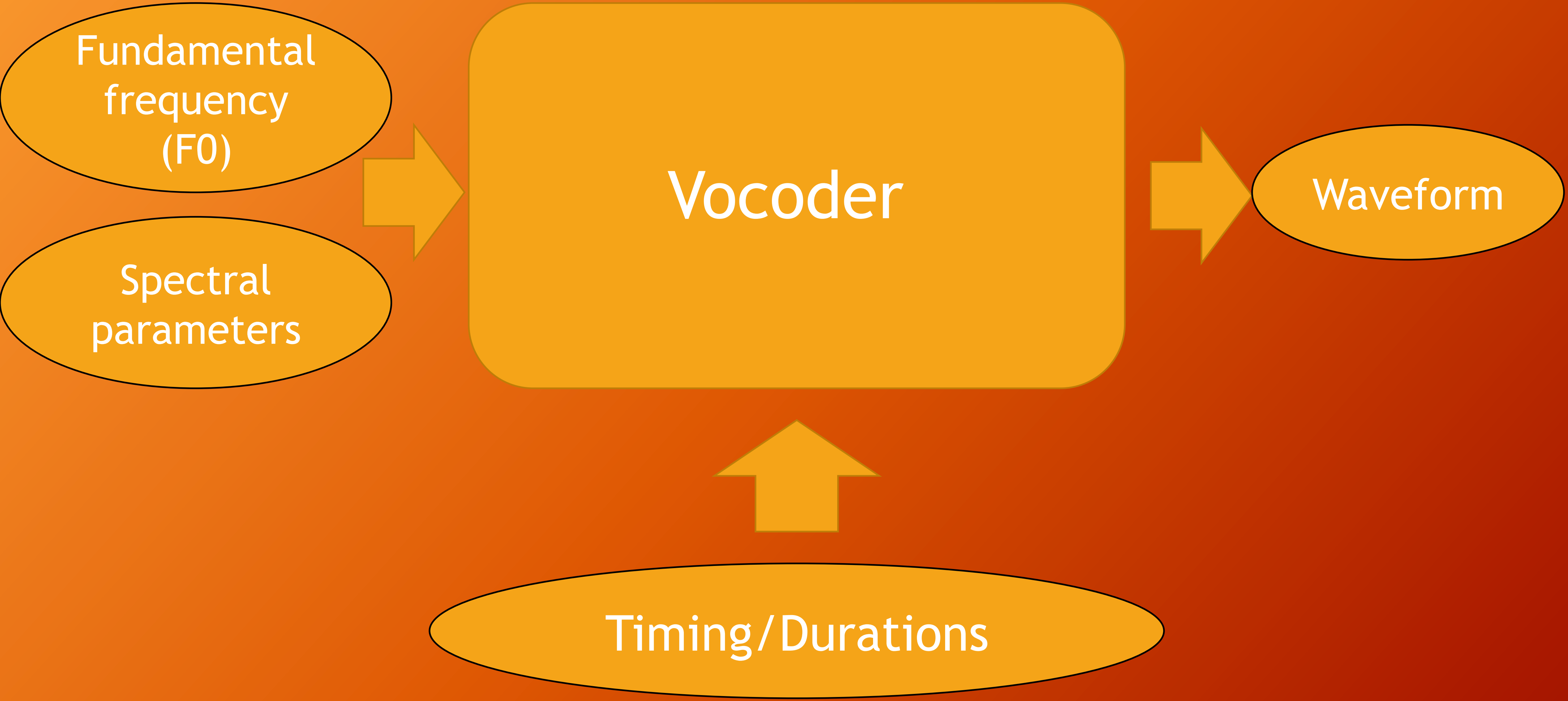
Speech technology

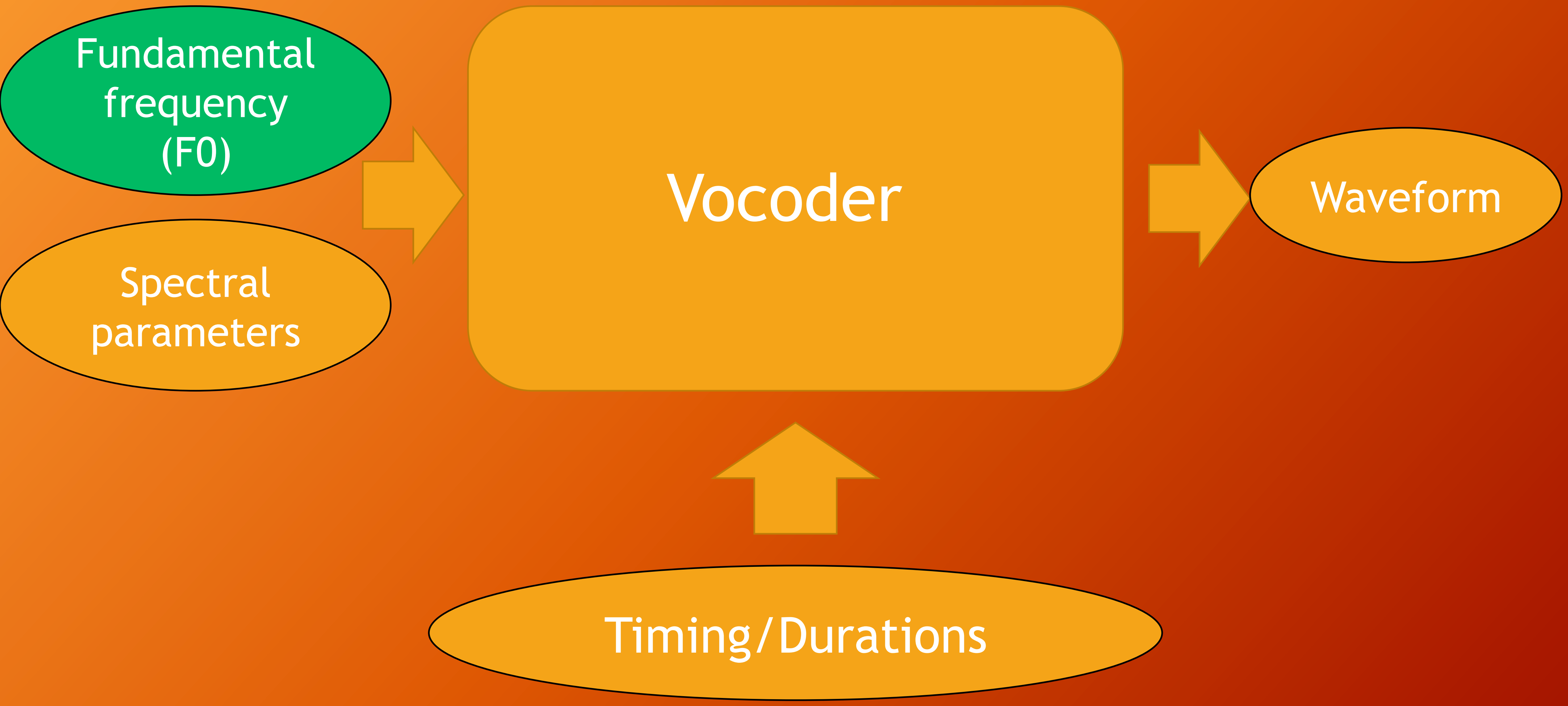


Text-to-Speech - fundamentals

- Written text -> Waveform (.wav)
- A popular approach for general purpose synthesis:

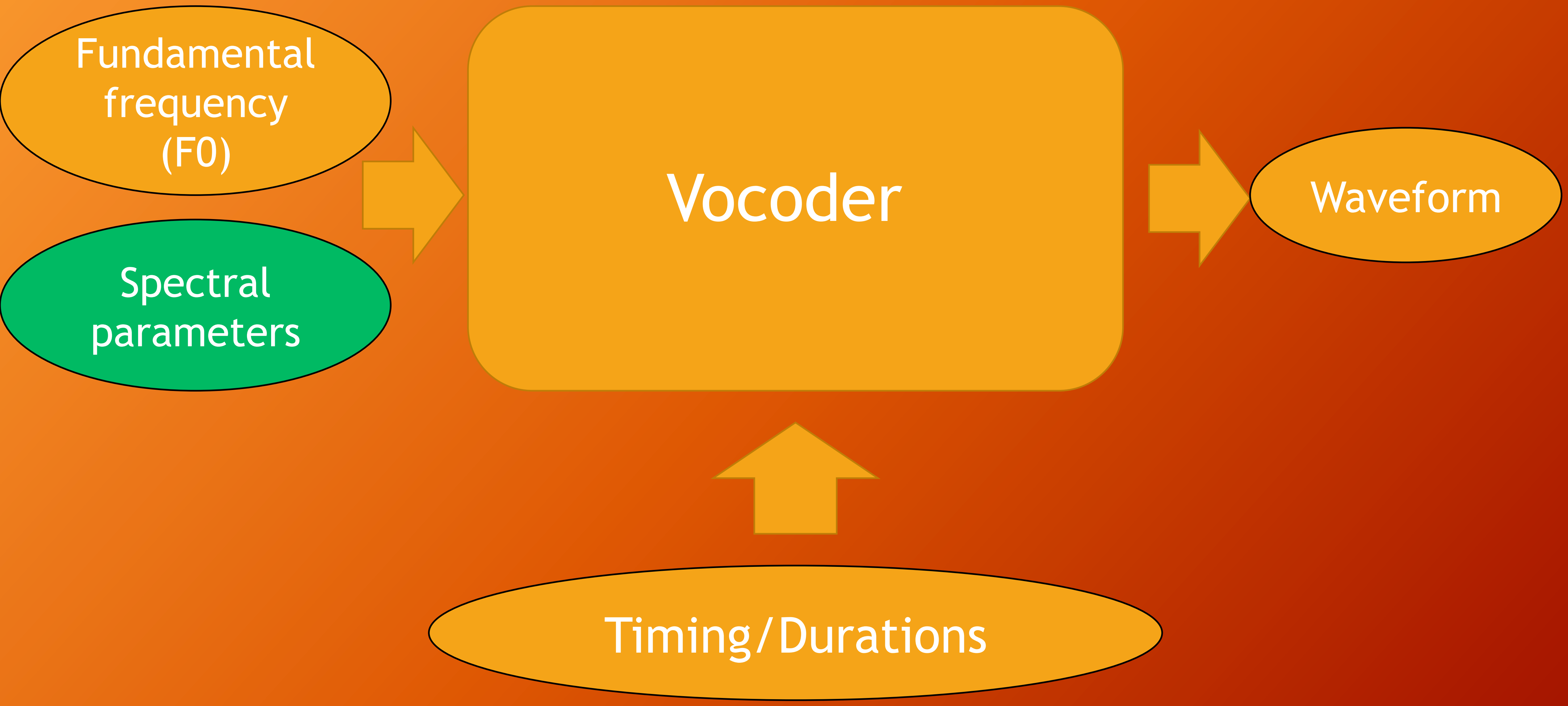
statistical parametric synthesis


Fundamental frequency (F0)
Spectral parameters
Vocoder
Waveform
Timing/Durations


Fundamental frequency (F0)
Spectral parameters
Vocoder
Waveform
Timing/Durations

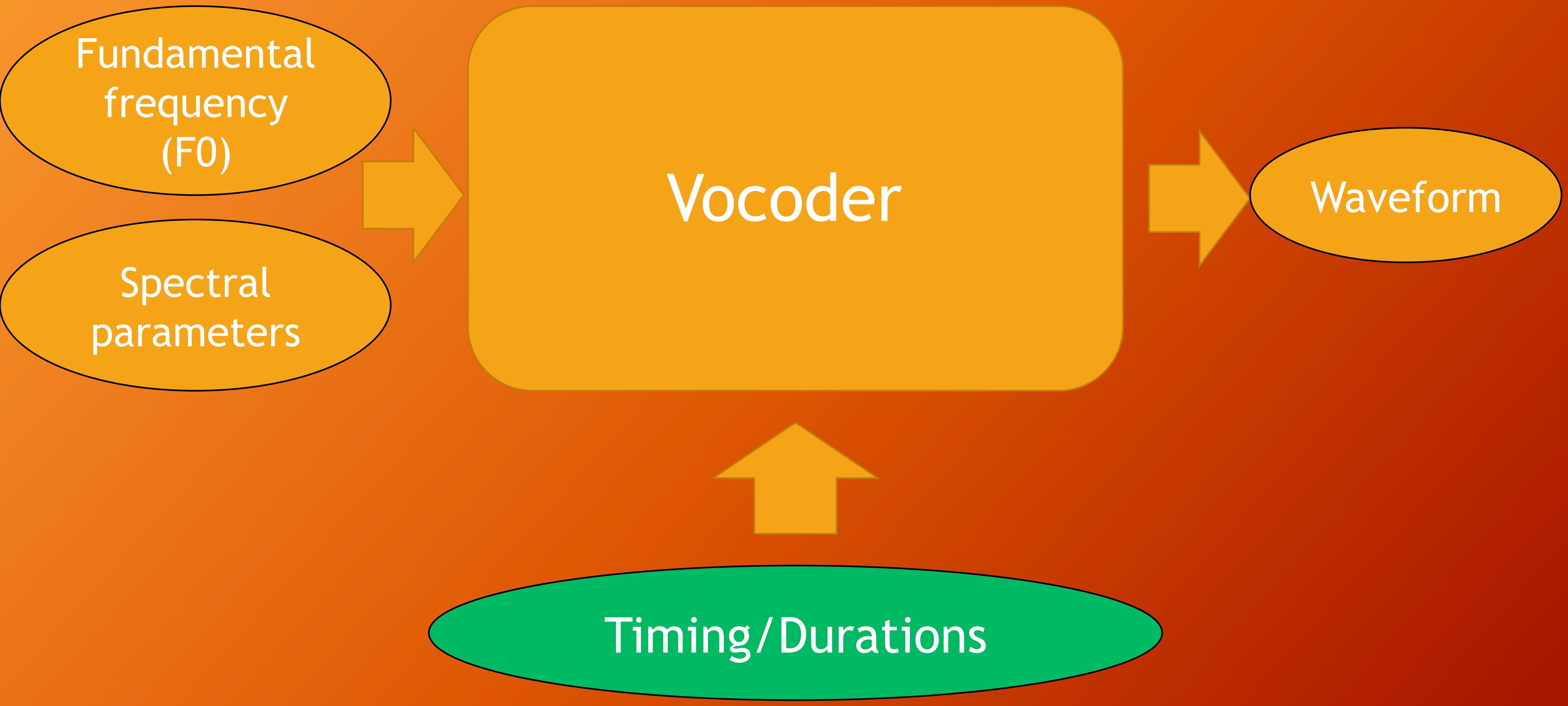
The fundamental frequency

- Different for all speakers
- Real-valued 1D function of time
- Discontinuous (!) - voiced/unvoiced flag + interpolation


Fundamental frequency (F0)
Spectral parameters
Vocoder
Waveform
Timing/Durations

Spectral parameters

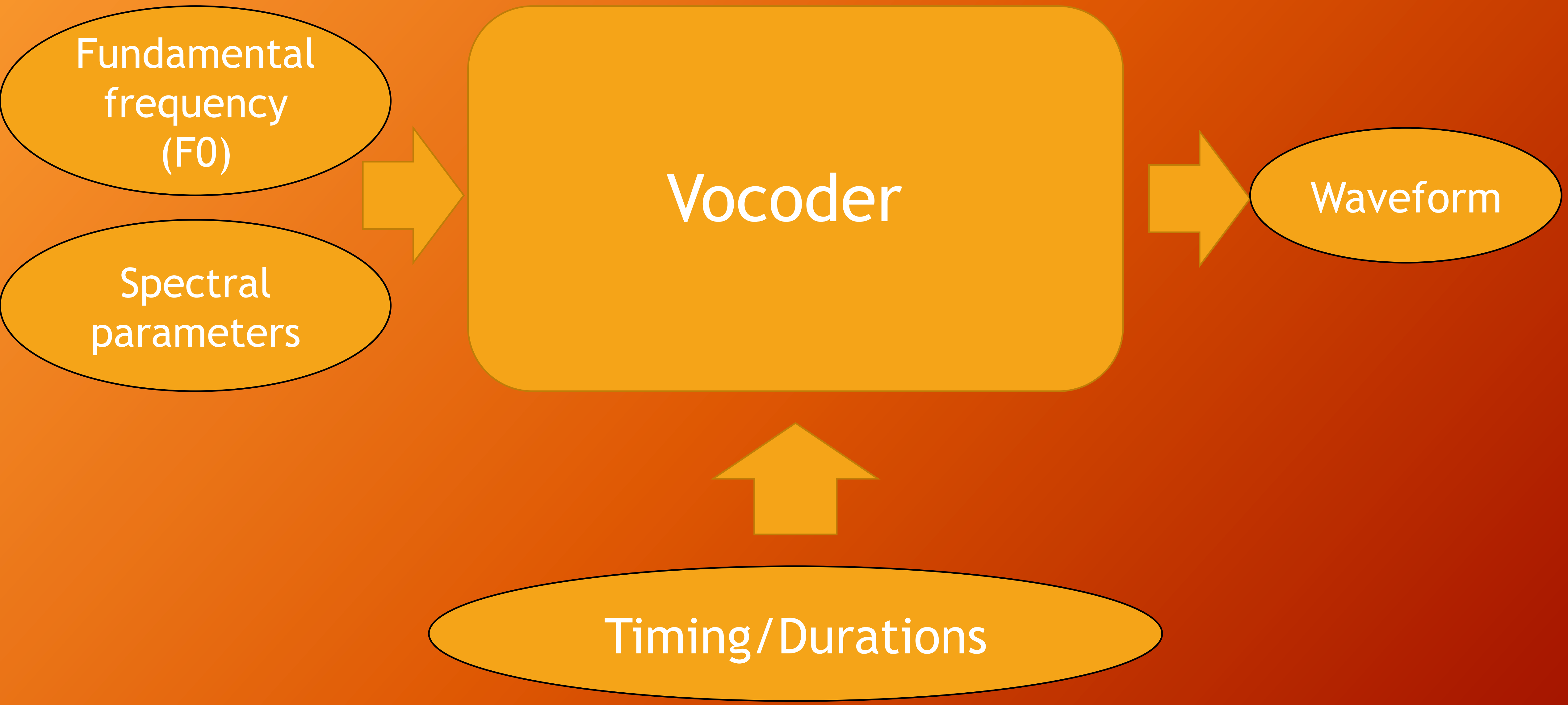
- LPC (Linear Predictive Coding) coefficients
  - Used to capture the information of speech on higher frequencies
  - Depth can vary: 12-48 (usual: 24+1 members.)


Fundamental frequency (F0)
Spectral parameters
Vocoder
Waveform
Timing/Durations

Timing/Duration

- When is the first/next phoneme starting?
- How 'long' is a phoneme?

- Has to be modeled on a different, smaller neural net


Fundamental frequency (F0)
Spectral parameters
Vocoder
Waveform
Timing/Durations

Database
~2 hours of recorded audio from a speaker (.wav files)
Short, declarative sentences

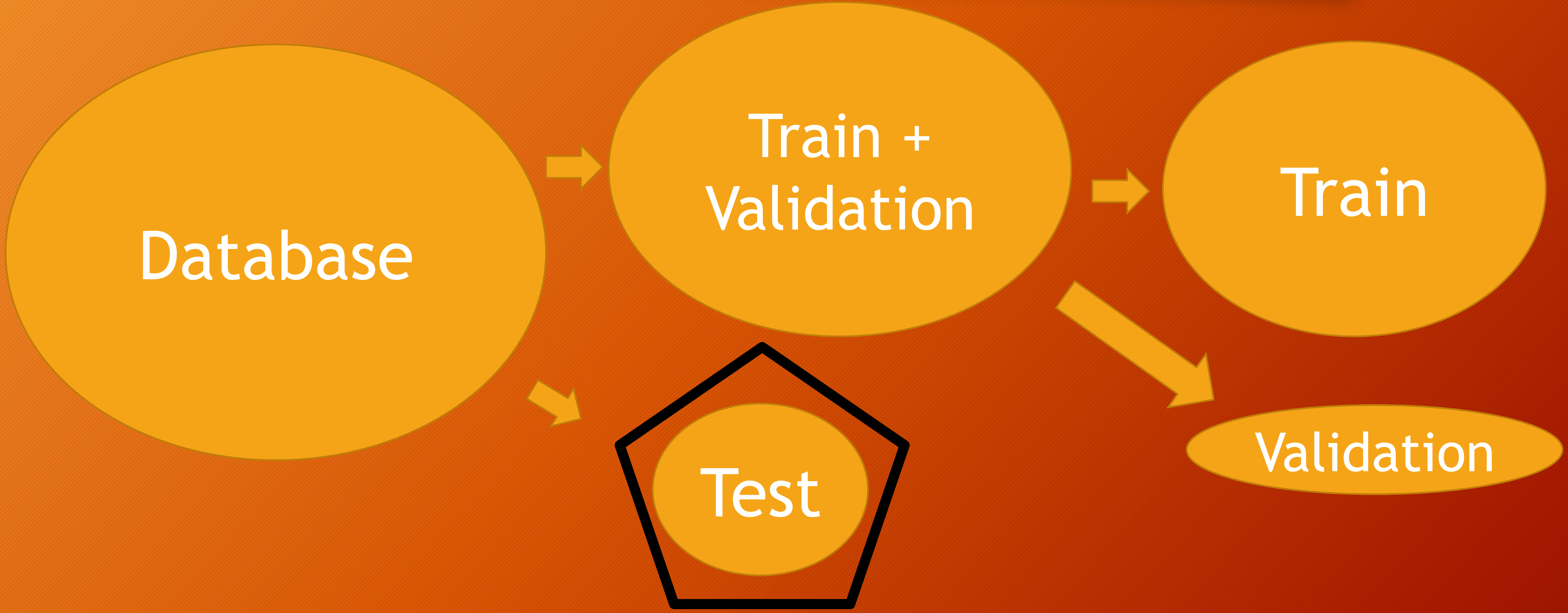
Phonetic transcripts

Neural network input
Statistical features

F0 extractor
LPC coder

F0 + LPC ground thruth

Database splitting

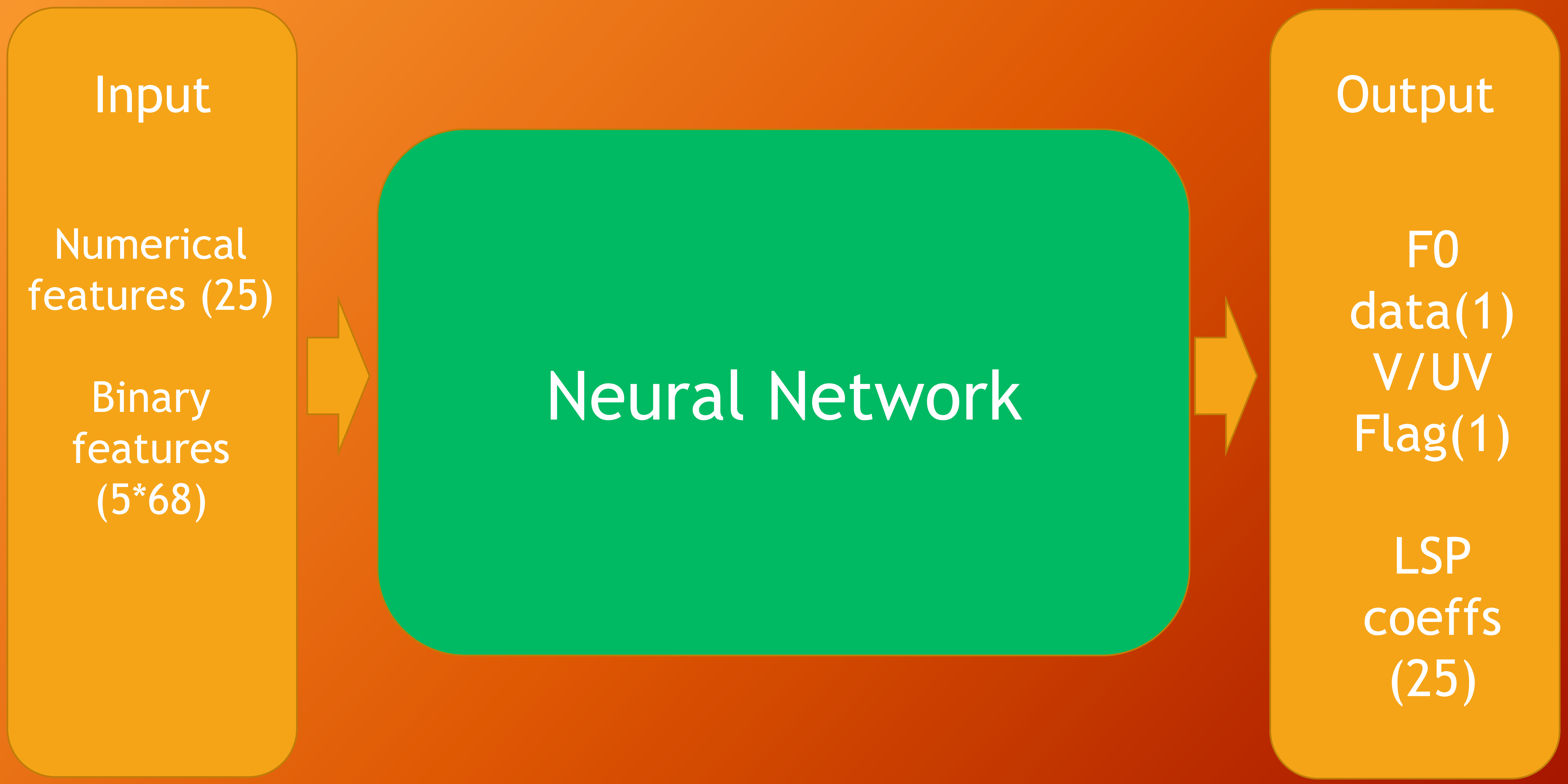



Input
Numerical features (25)
Binary features (5*68)
Neural Network
Output
F0 data(1)
V/UV Flag(1)
LSP coeffs (25)


Input
Numerical features (25)
Binary features (5*68)
Neural Network
Output
F0 data(1)
V/UV Flag(1)
LSP coeffs (25)

The input

- Binary features:
  - ’Quinphone’ model (first and second neighbours in phonemes)
  - 5 * 68 (Hungarian) = 340
- Numerical features – representing a broader context:
  - Examples:
    - Total number of phonemes in the current word/sentence
    - Duration of current phoneme (ms)
    - Prosodic stress level (sentence, word, phoneme etc.)


Input
Numerical features (25)
Binary features (5*68)
Neural Network
Output
F0 data(1)
V/UV Flag(1)
LSP coeffs (25)

The neural networks

- Multi-Layer-Perceptron (MLP)
  - Stack of FC layers
  - Feedforward

- Alternative: LSTM, GRU etc. -> recurrent nets
  - Or CNN (very recent result -> WaveNet)

Why MLPs?

Deep MLPs
(feedforward)

New idea:
Backprop.
through
time

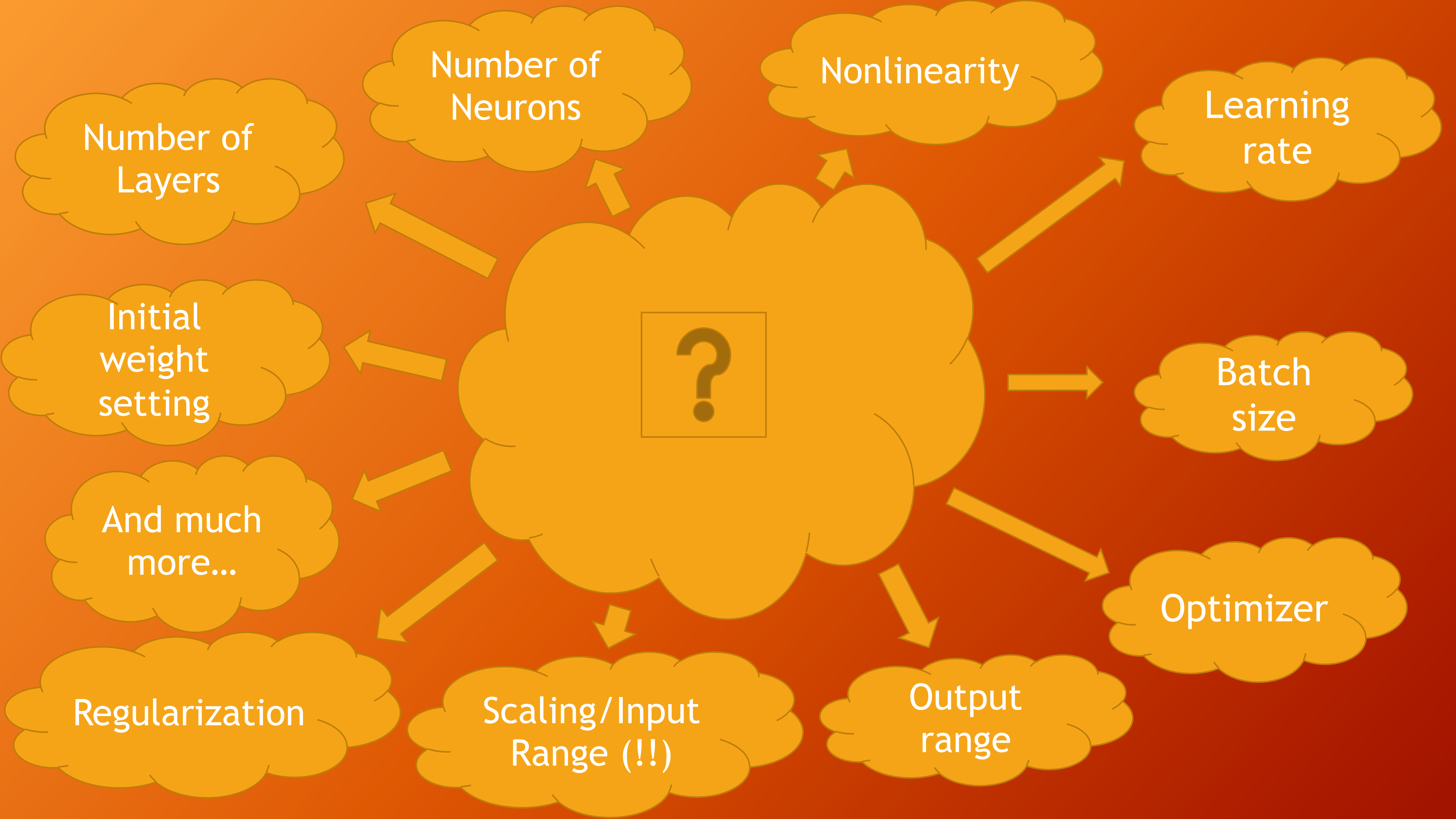
LSTM, GRU
etc.
(recurrent)

New idea:
Input is
highly
redundant

ConvNETs
(feedforward)

Things to consider…

- Number of Layers
- Number of Neurons
- Nonlinearity
- Learning rate
- Batch size
- Optimizer
- Output range
- Scaling/Input Range (!!)
- Regularization
- Initial weight setting
- And much more…


Number of Layers
Number of Neurons
Nonlinearity
Learning rate
Initial weight setting
Batch size
And much more…
Optimizer
Regularization
Scaling/Input Range (!!)
Output range

Number of Layers

- More than one ☺ ->’deep learning’
- Type: integer
- ’usual’ value: 2-6 (MLP)

- >10 -> vanishing gradients problem

Number of Neurons

- Fundamental unit of the network (weights + bias)

- Type: integer
- ’usual’ value: 2- (few thousands)

- Training time – performance tradeoff
- Same number in each layer -> usually better
- Too many -> overfitting, too few – bad performance

Non-Linear function

- Output = f(W*x + b), x is the input

- f(…) is the nonlinearity
- Usually: ’ReLU’ or something similar

- Sigmoid, tanh, -> not really…

Learning rate

- Probably the most important hyperparameter
- x +=   - learning_rate *dx
- manual setting -> minibatch gradient descent
- Type: float (0.0-1.0)
- ’usual’ value: 1e-5 – 0.3 (no guarantees ☺)
- Larger nets -> usually smaller

Batch size

- Can be tricky…
- HUGE training time – performance tradeoff
- Minibatch learning -> approximating the real gradient

Optimizer

- ’Classic’: Mini-batch with momentum
- ’Classic+’: Mini-batch with Nesterov momentum

- Novel methods: RMSprop, Adagrad, Adadelta, Adam, Nadam …

Input range/output range

- problem-dependent
- Should be close to the range of network weights
- Proper scaling of input is essential
  - Do NOT fit scaling on the test database !

Regularization

- Weapons against overfitting
- Early stopping: recommended
- Dropout:
  - Type: float
  - 'Usual' values: 0.1-0.5
  - Very useful
- L1 and L2 regularization
  - L1 ~ a bit like PCA
  - L2 = weight decay

Initial Weight settings

- Some form of random initalization
- Examples: unform, orthogonal, lecun_uniform, glorot_uniform etc.
- Better choice may speed up the learning process

- More about this here: http://yann.lecun.com/exdb/publis/pdf/lecun-98b.pdf
- Yann LeCun et al. : Efficient Backprop

Some hints for training

- Training error / valid error should be close -> if training goes down, valid not -> overfitting
- Do not use the test database in any way during training !
- Always make separate evaluation of the results (not just the error rate)
- Use a fast GPU (or several fast GPUs…)

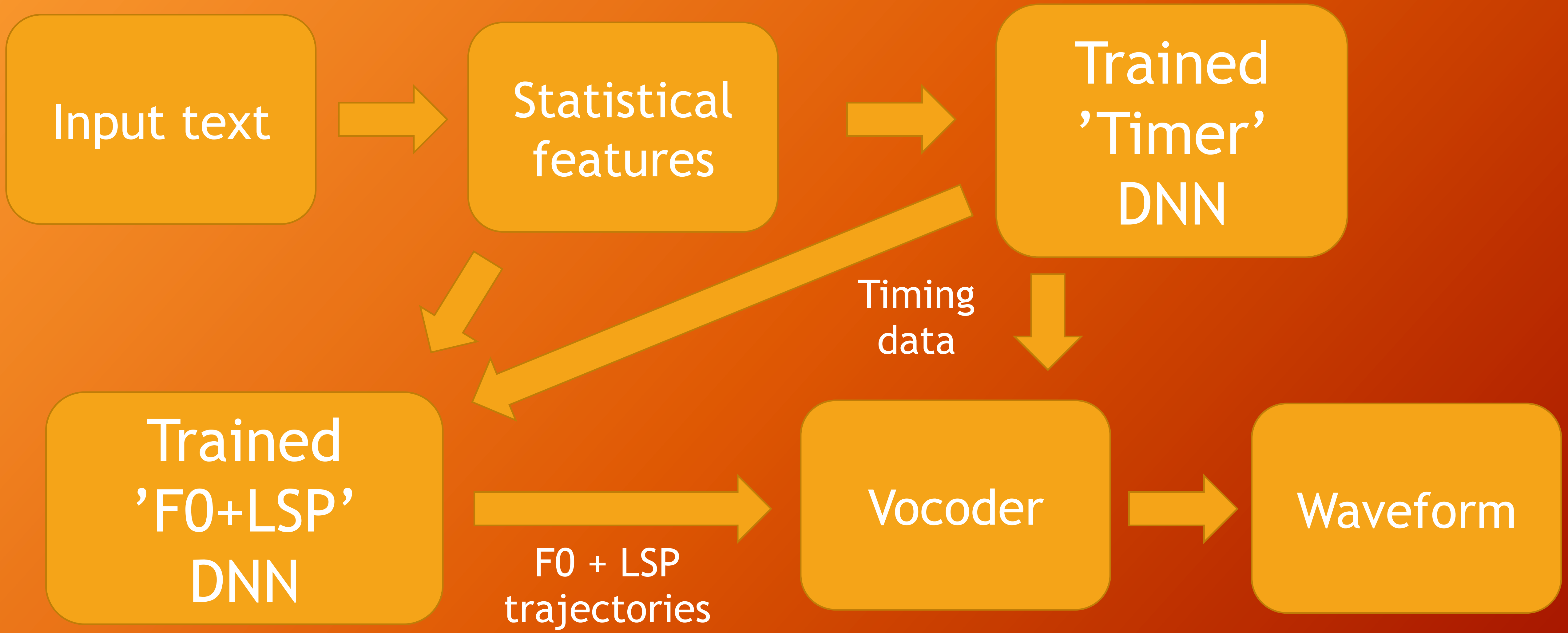
Hyperparameter optimization

- Search in the space of hyperparameters
  - Usual dimensions: learning rate, batch size, neuron number, layer number
- Approaches: Manual Search, Grid Search, Random Search, etc.

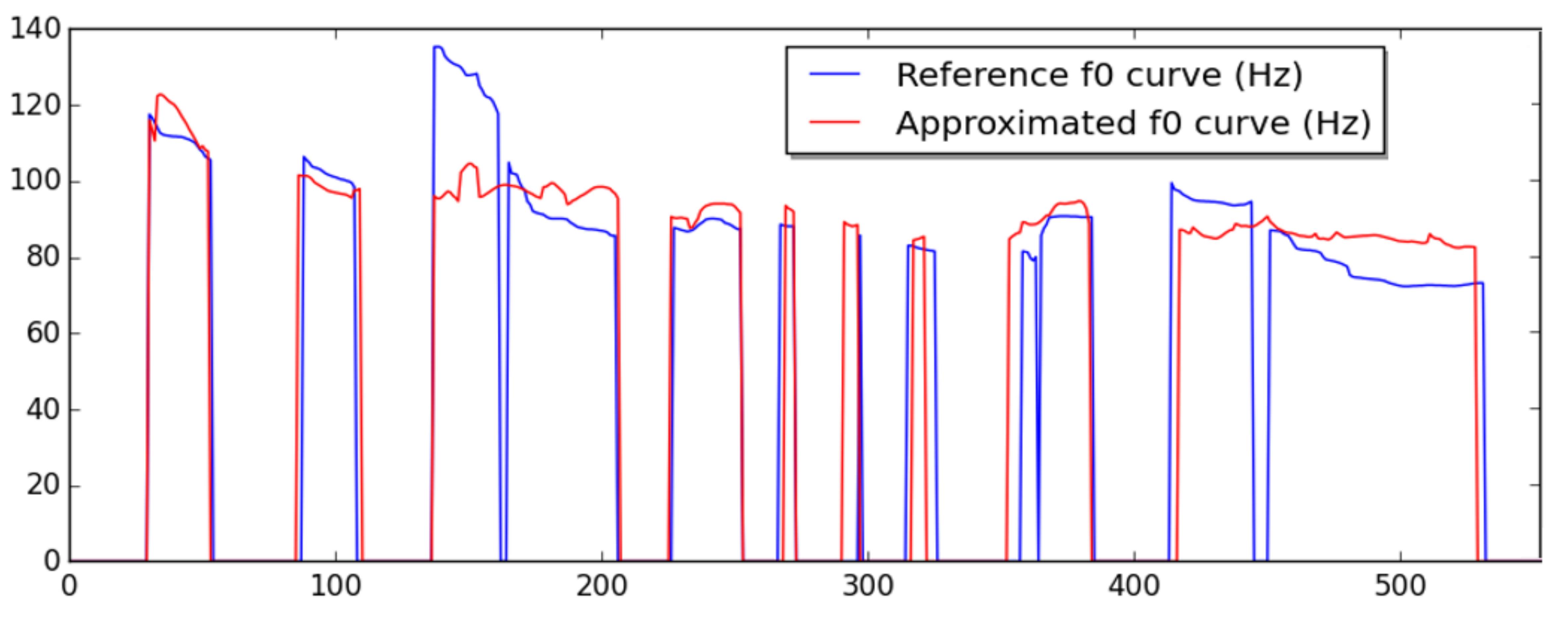
Hyperparameter optimization

- Manual search: not very effective
- Grid search: good, but very time-consuming
- Random search: probably the best

The complete DNN-TTS (testing)



Results



Reference f0 curve (Hz)
Approximated f0 curve (Hz)
0
20
40
60
80
100
120
140
0
100
200
300
400
500

Ensemble learning

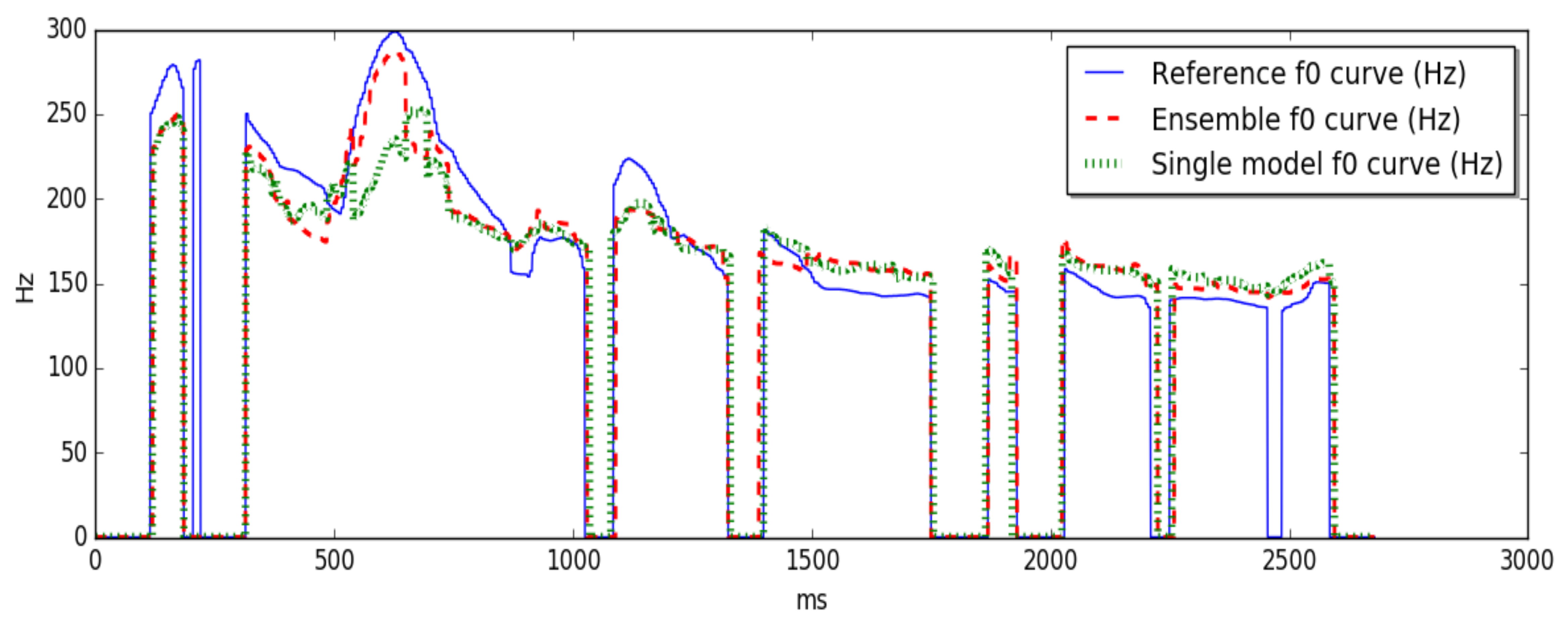
- Idea: Group of specialists OR group of weaker members working together

- Good way to get a little bit better results from our system
- The members does not have to be neural networks!

Ensemble in speech technology (our idea)

- Train individual nets on different parts of database
  - The decision point the the level of prosodic stress
- Each member is responsible for one level only
- Goal: Better stress estimation

Results



Reference f0 curve (Hz)
Ensemble f0 curve (Hz)
Single model f0 curve (Hz)
Hz
ms
0
50
100
150
200
250
300
500
1000
1500
2000
2500
3000

Thank you for your attention!

Questions are welcome