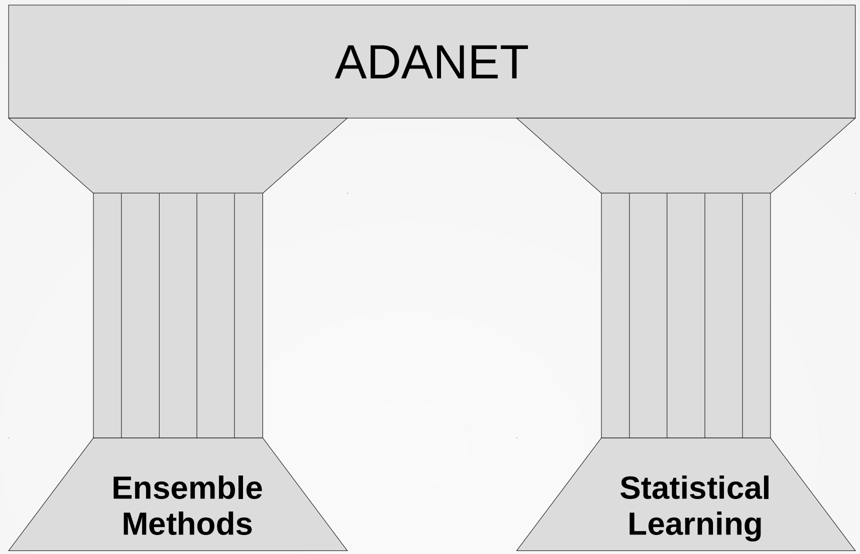
Parcours Datascientist : projet 8

AdaNet
Méthode ensembliste
appliquée aux réseaux DNN, CNN, RNN

François BANGUI

Adanet : theoritical frame



- Ensemble methods applied to N.N.
- Build an ensemble N.N from sub-networks
- Each sub-network is a weak learner

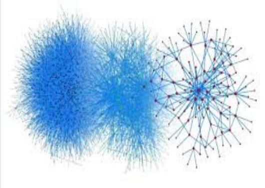
Deep Boosting (2014) extension

- Empirical Risk Minimization Principle
- Structural Risk Minimization Principle
- Complexity of family of classifiers

Vapnik-Chervonenkis /
Dimension VC (1999)
Koltchinskii : Rademacher (2001)

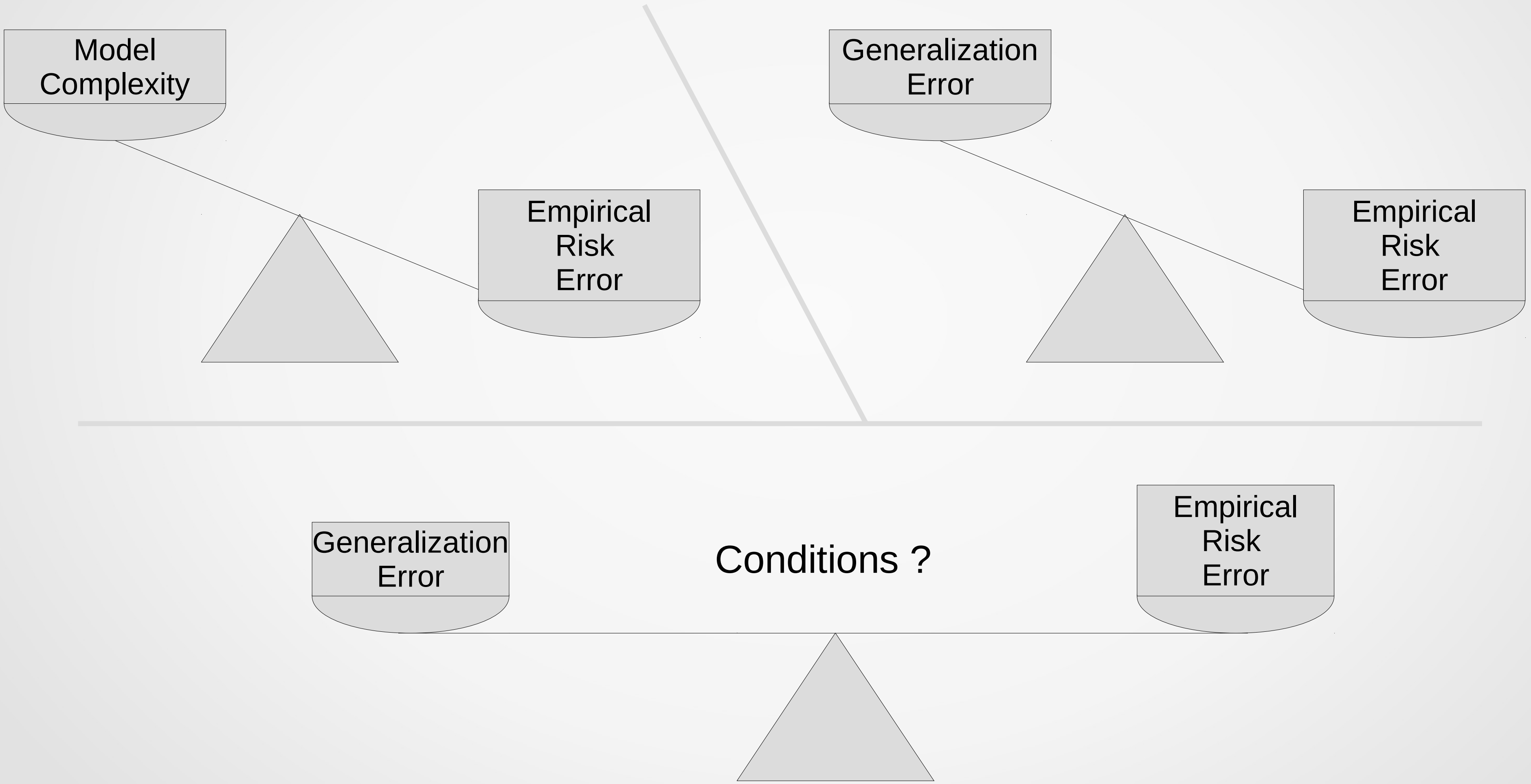
Adanet : modeling complexity

- What is complexity ?
  - Complex problem : number of informations to « describe » it
  - Complex model ↔ Number of elements in model capturing information
  - For a N.N elements are:
    - Number of layers
    - Number of units in each layer
    - But also :
      - Number of type of layers in model
      - ….



- What are complexity issues?
  - Complex problem ⇒ Complex model
    - Low empirical error → Trained Dataset
    - Low generalization performance → Test Dataset

Adanet : building a « consistant » model

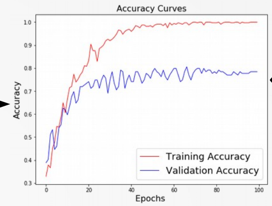


Adanet : Risk Minimization Principles

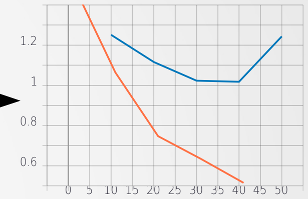
Empirical Risk Minimization Principle :

- Decrease empirical error when building model
- Decreasing such risk ⇒ Expectation of good Generalization, whatever ?

NO : $\hat{\mathfrak{R}}(f, S) \leqslant \mathfrak{R}(f)$



« worst »



When can it be expected :

- ERM decrease ⇒ Learning guarantee?

$\mathfrak{R}(f)$: *Generalization Error*

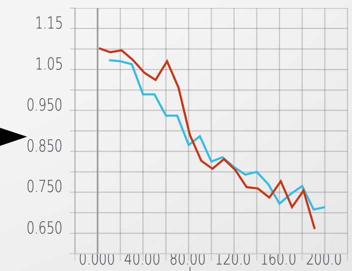
$\hat{\mathfrak{R}}(f, S)$: *Empirical Risk*

$f \in F$: *learned function*

S : *trained dataset*

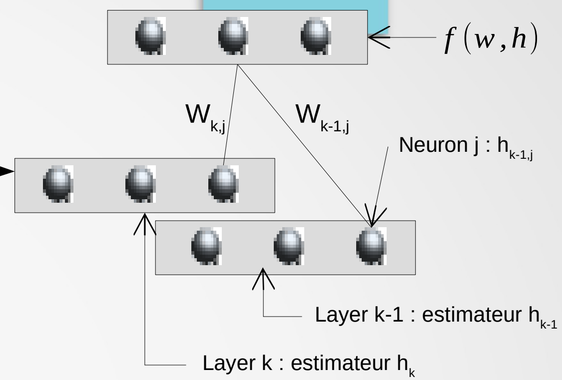
Consistancy of ERM:

- $\mathfrak{R}(f) \leqslant \hat{\mathfrak{R}}(f, S) + C(F)$
- F : *family of estimators, with* : $f \in F$
- C : *measures complexity of* F



Structural Risk Minimization Principale

AdaNet : structural adaptative learning



$$f(w,h) = \sum_{j}^{N} w_j h_j$$

Structural :

- $\boldsymbol{h_j}$: jth Layer of network f
- $\boldsymbol{w_j}$: mixture weights : matrix, vector or scalar

Cost function : adaptative learning

- $f(w,h) = \sum_{j}^{N} w_j h_j$

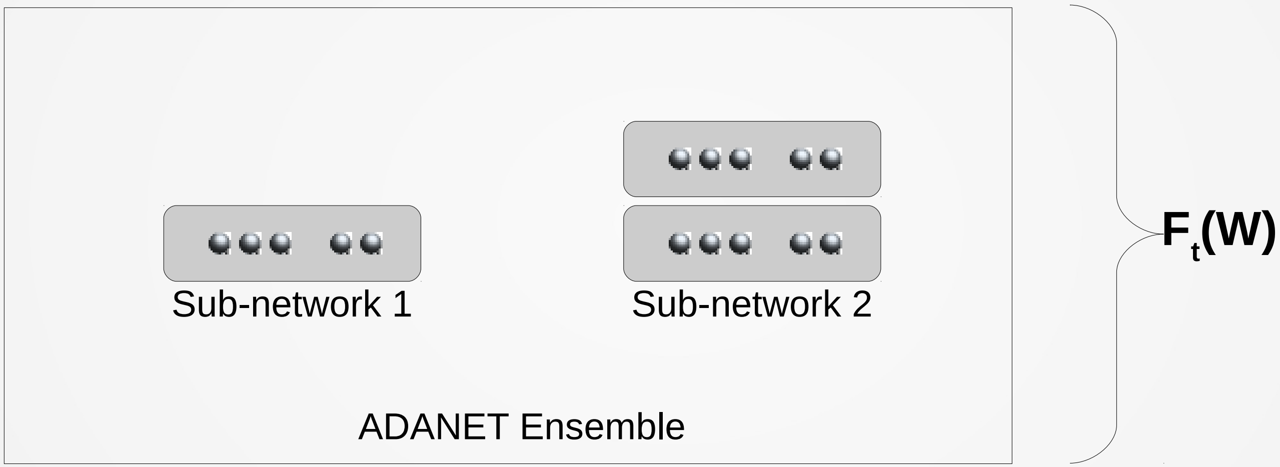
$$\Re(f) \leqslant \hat{\Re}(f,S) + C(F)$$

$$\Re(f) \leqslant F_t(w,x_i) = \frac{1}{m} \sum_{i=1}^{m} \Phi(1 - y_i . f_t(x_i, w_i)) + \sum_{j=1}^{N} \Gamma_j \|W_j\|_1$$

W_j *: mixture weights*
Γ_j *: Rademacher complexity of* f_j

Descent algorithm : block coordonates descent (boosting algorithms)

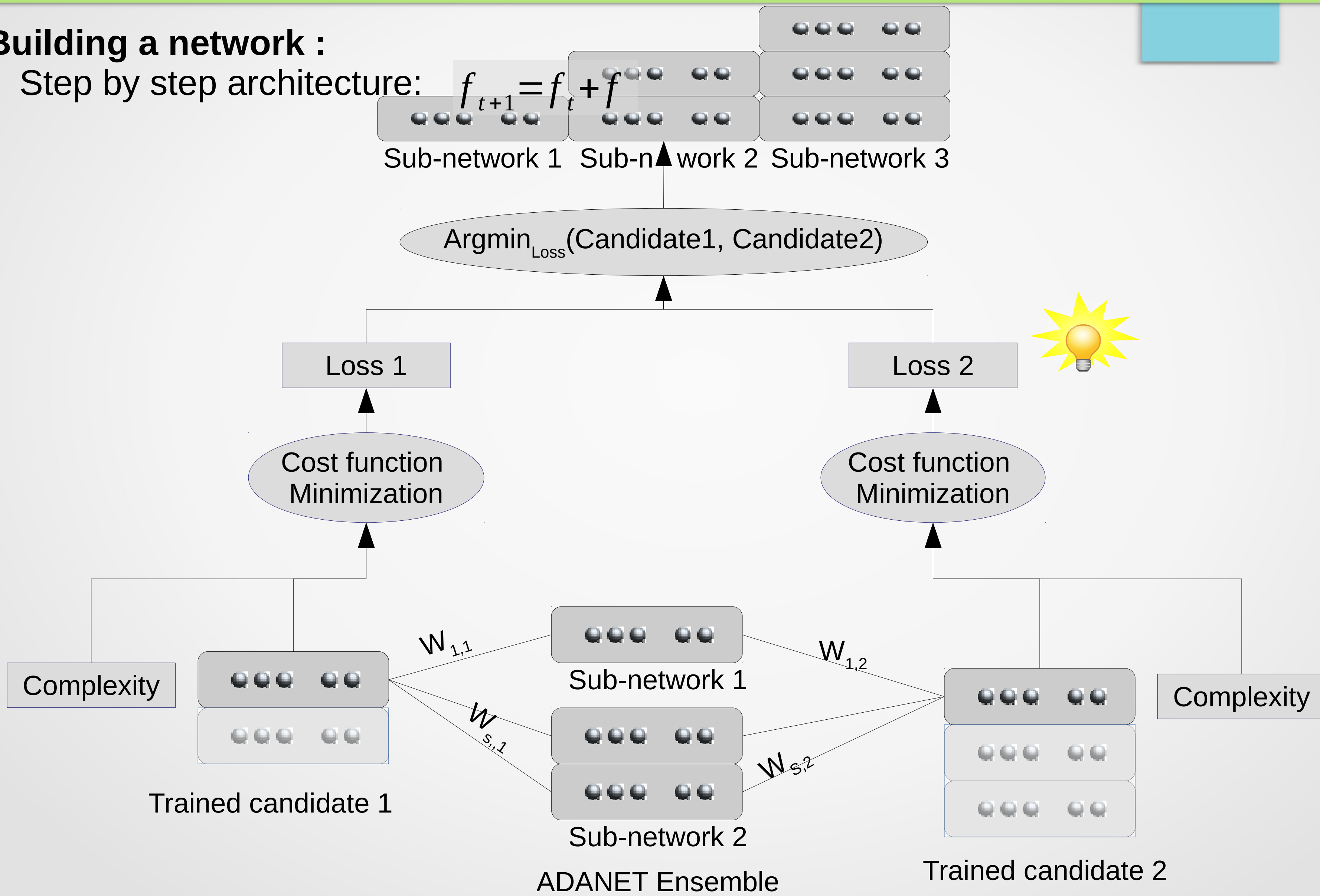
AdaNet : Algorithm weak learners (1)



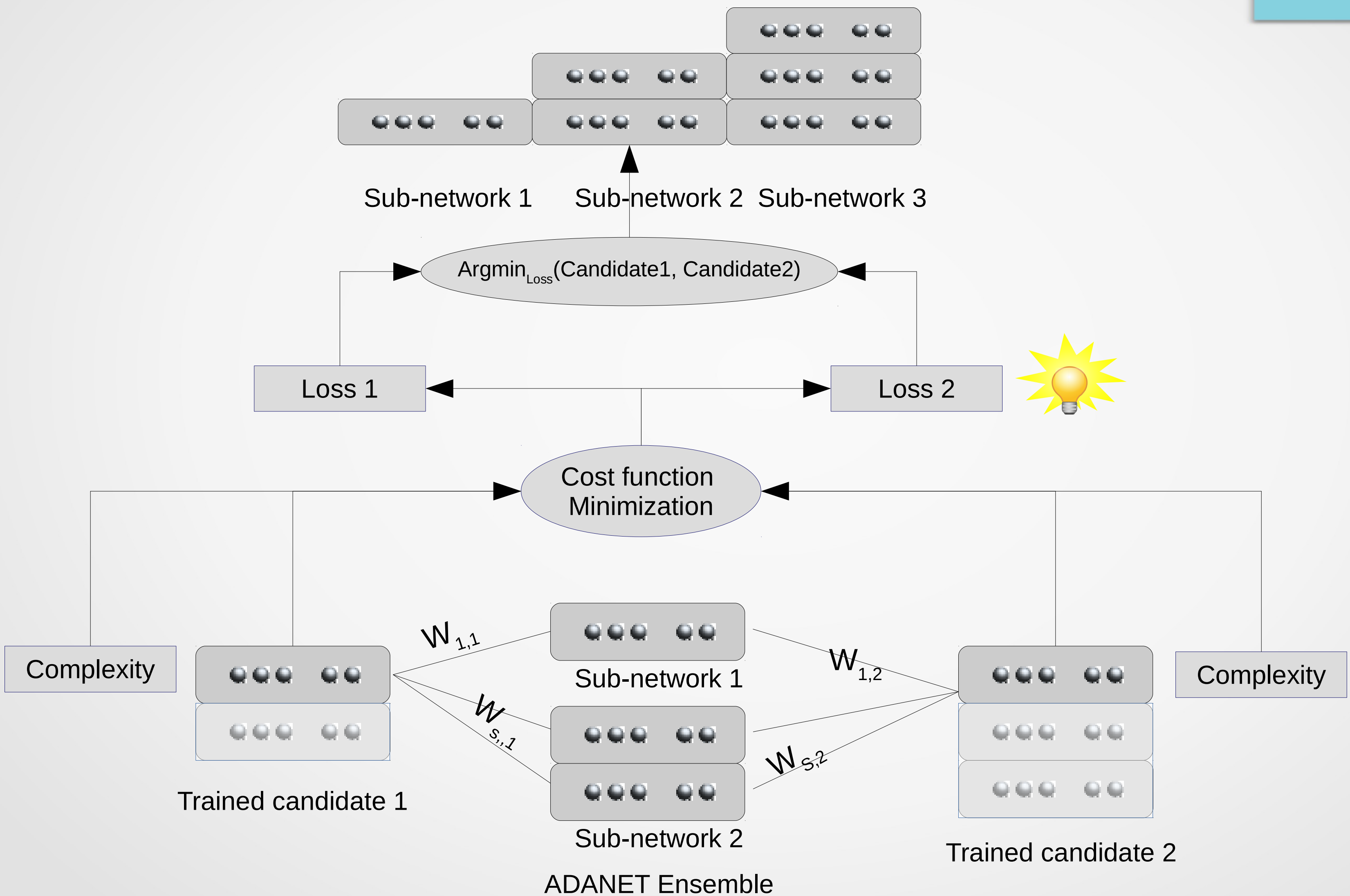
AdaNet : Algorithm weak learners

Building a network :

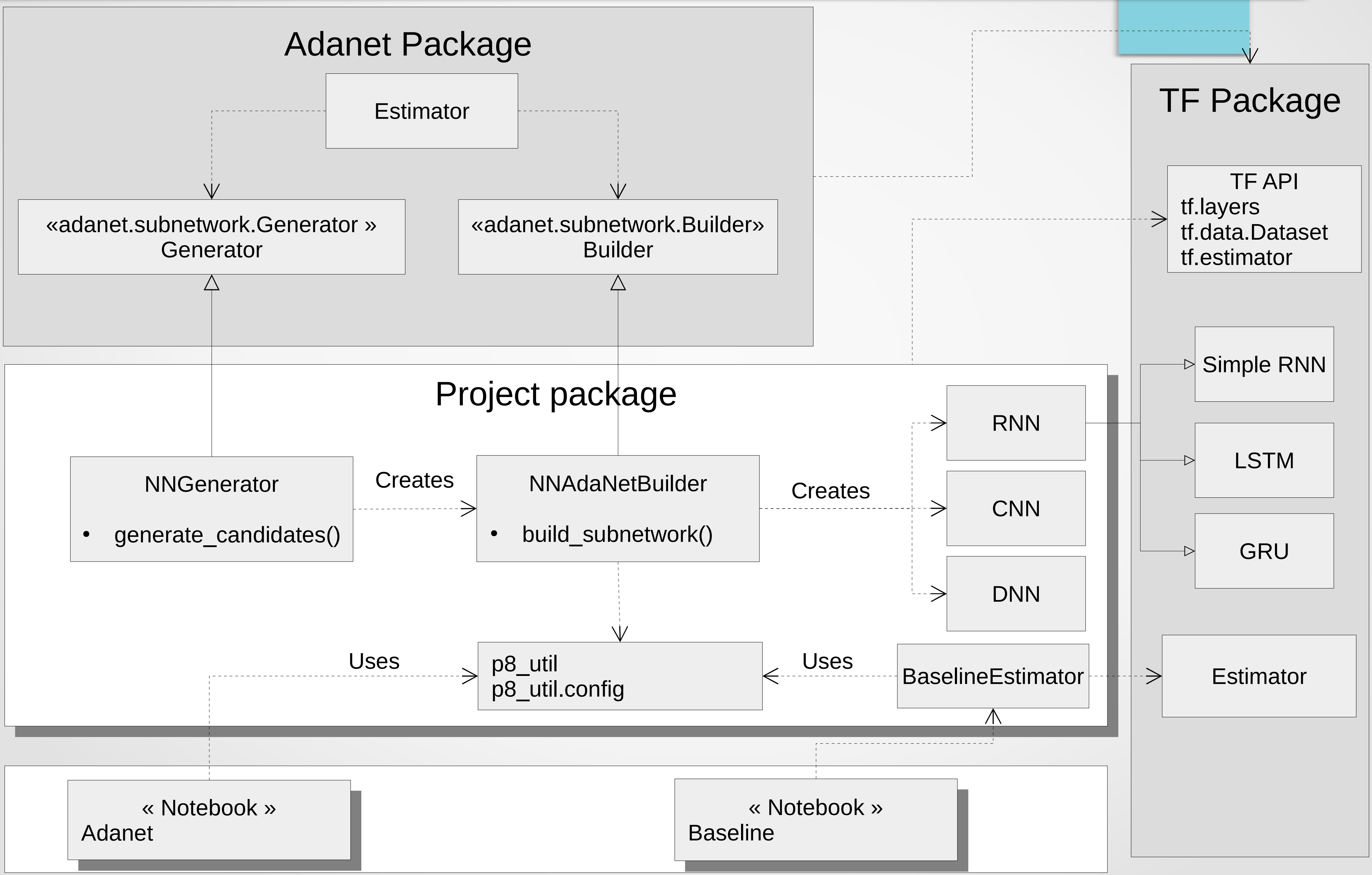
- Step by step architecture: $f_{t+1} = f_t + f$



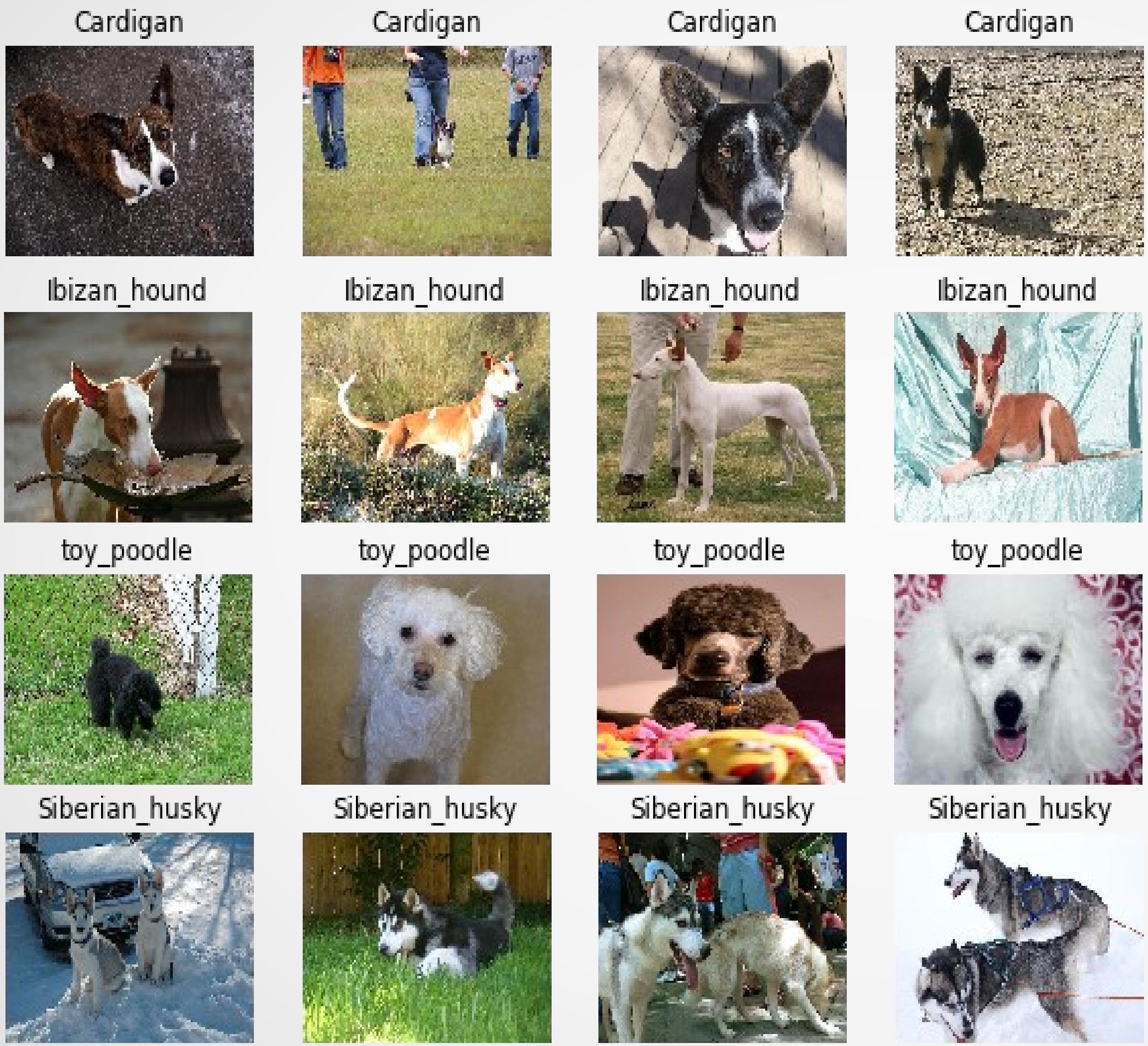
AdaNet : Algorithm weak learners



AdaNet implementation

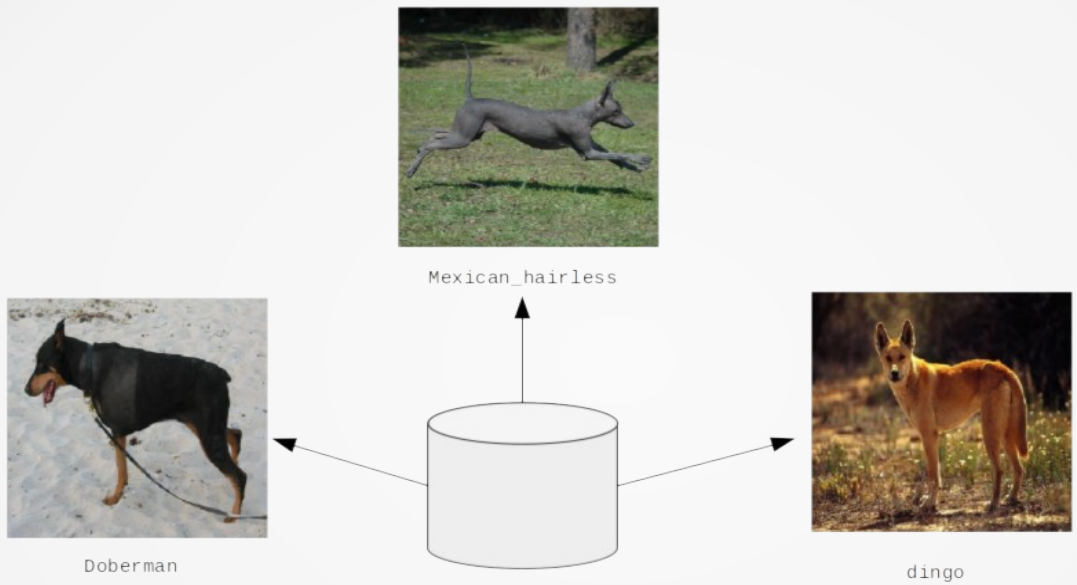


Images complexity : classification issues



- Backgrounds
- Topics confusion
- High variance : colors, profiles, sizes...
- Multiple subjects per image

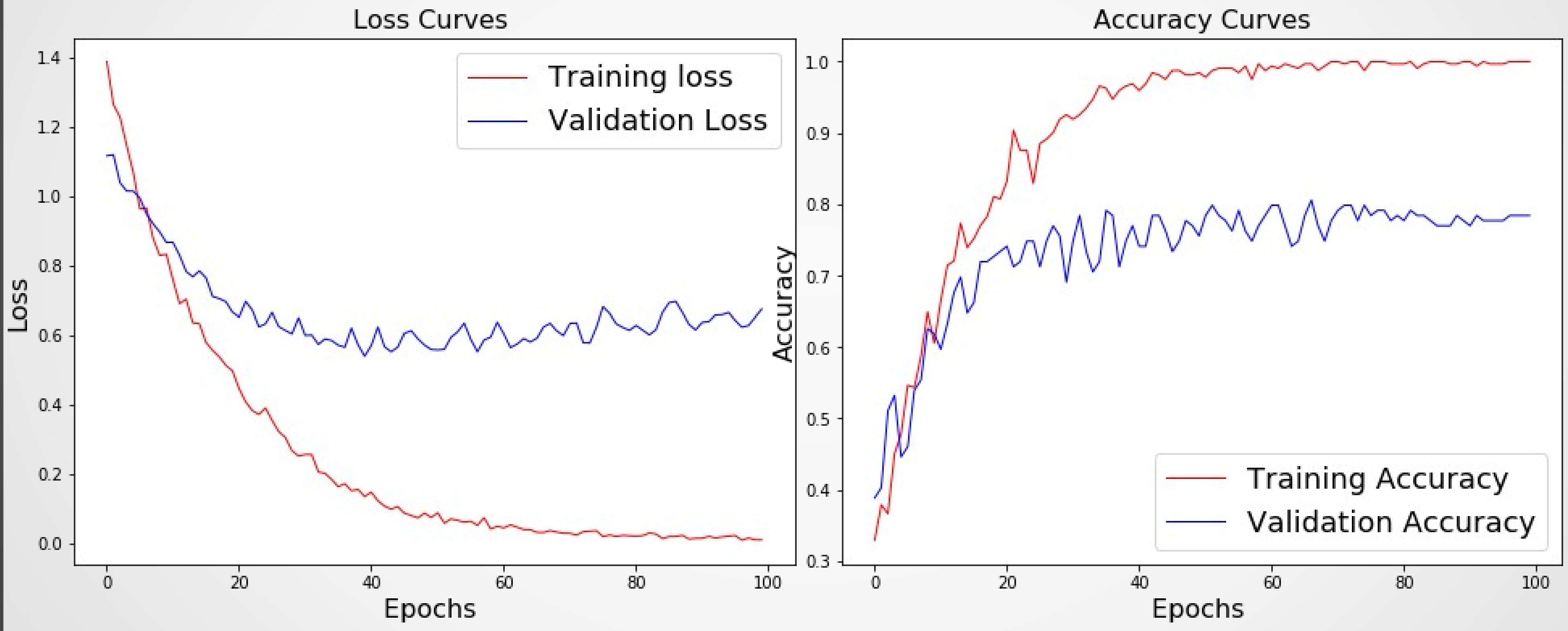
Dataset : 3 breeds



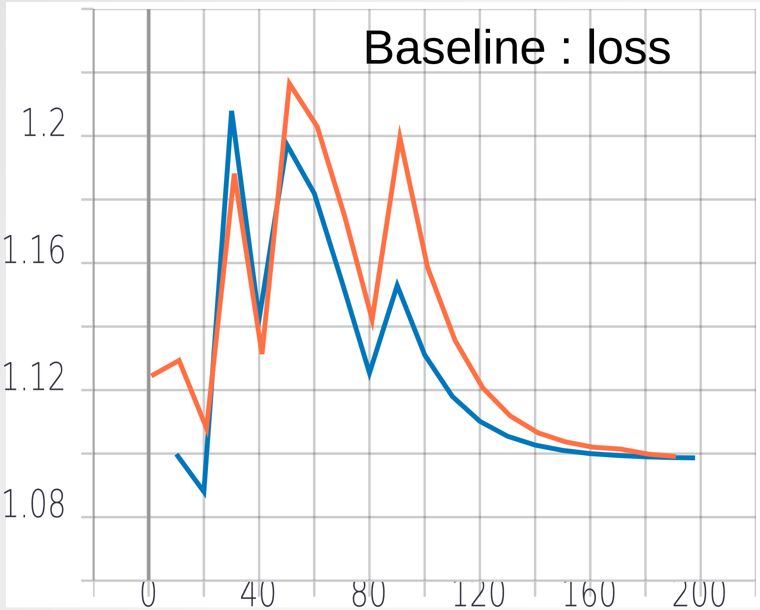
Standford Dogs Database

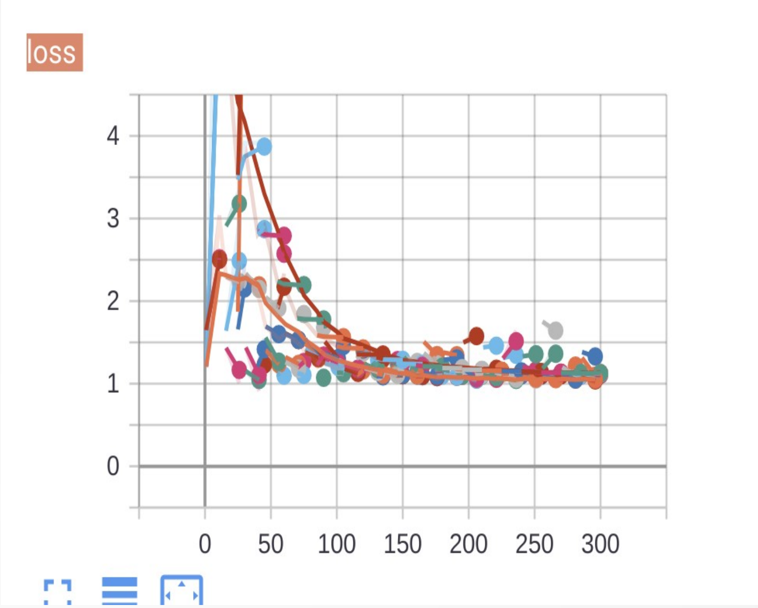
3 breeds
Train dataset : 400 images for 3 breeds
Test dataset: 40 images for 3 breeds

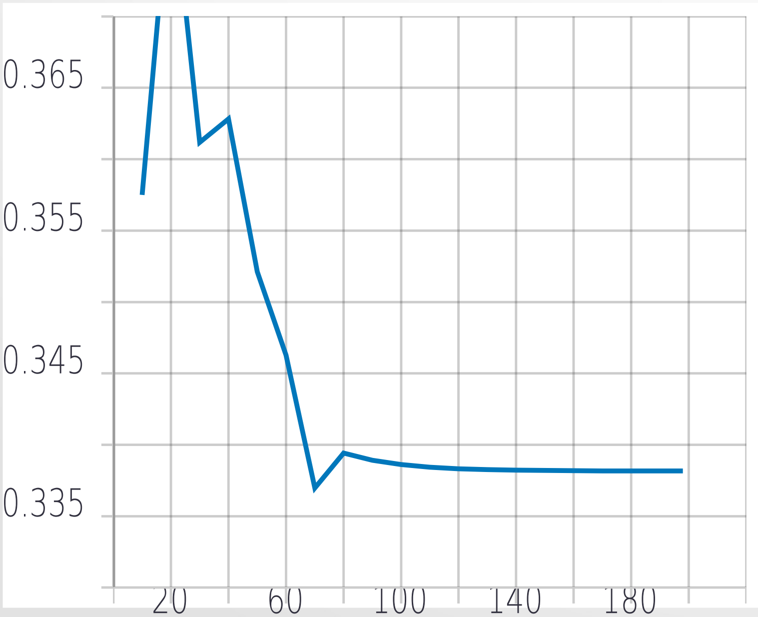
Reference: VGG-16 pre-trained

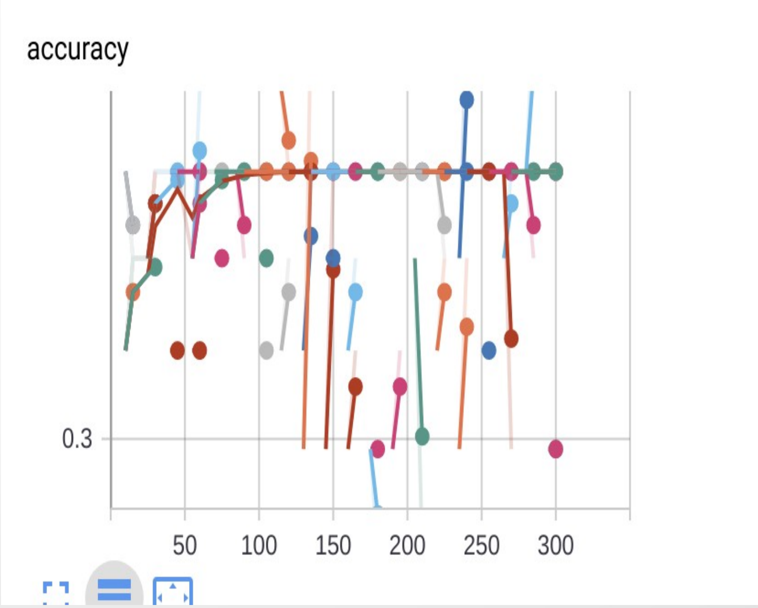


Adanet : DNN performances

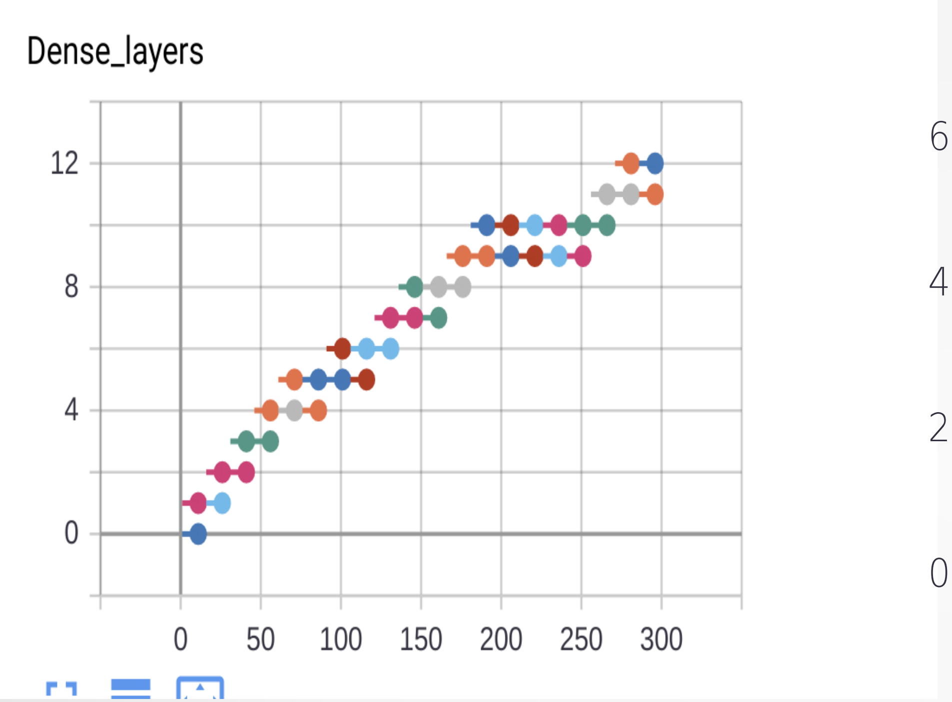


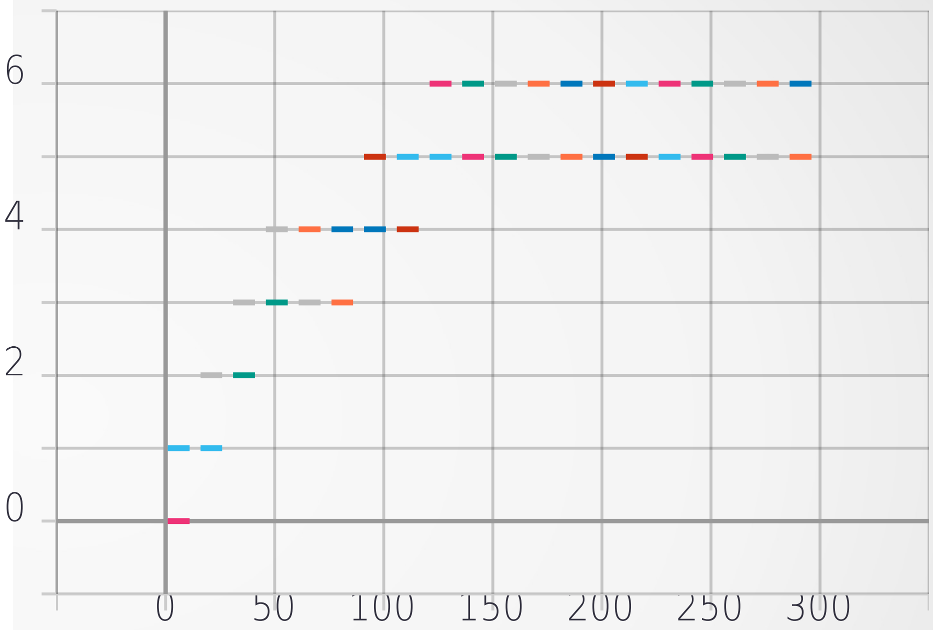




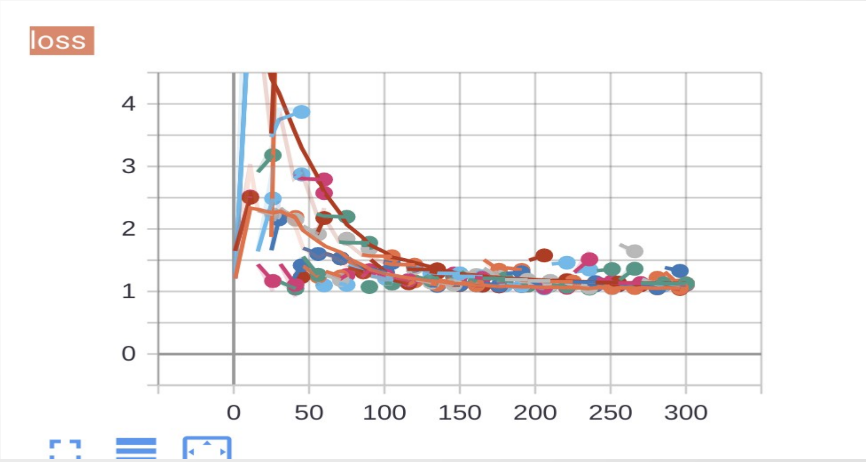


Adanet : complexity impact (DNN)

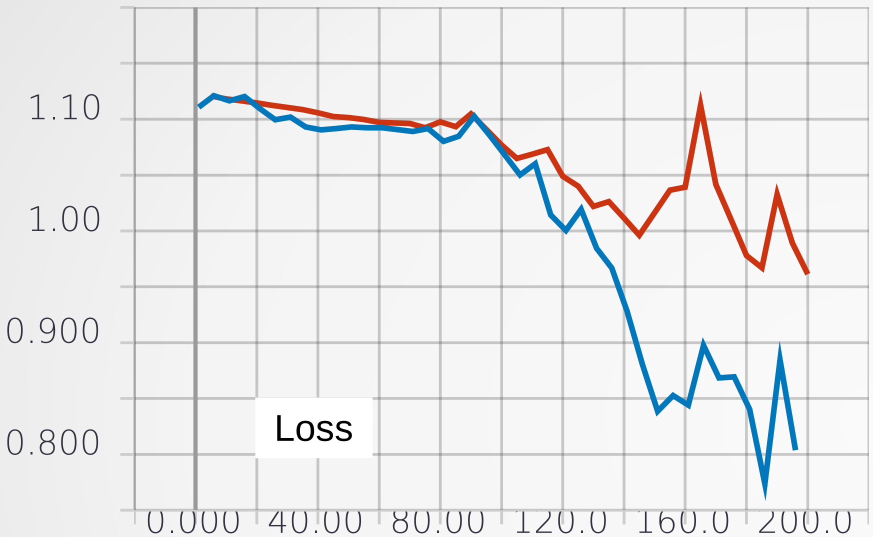


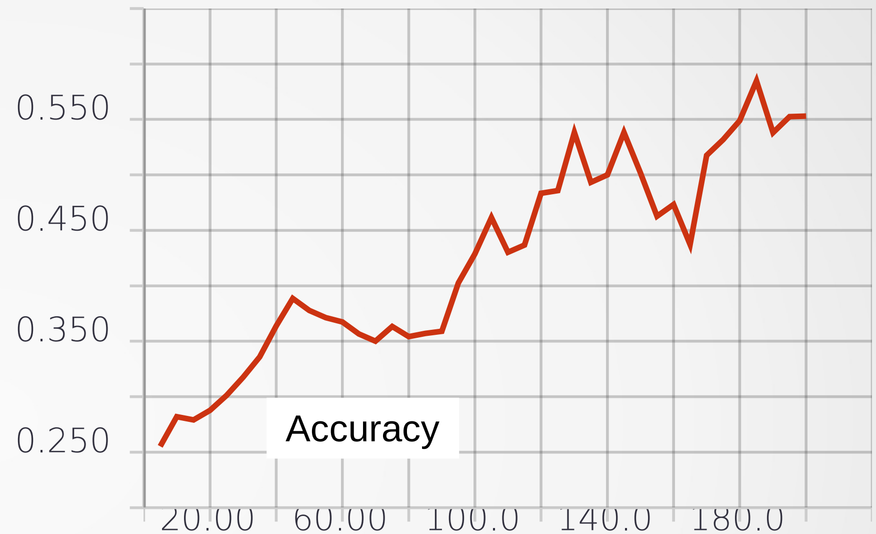


DNN network
Dense layers : 12
Steps : 300
ADANET iterations : 10
EVALAccuracy : 34 %



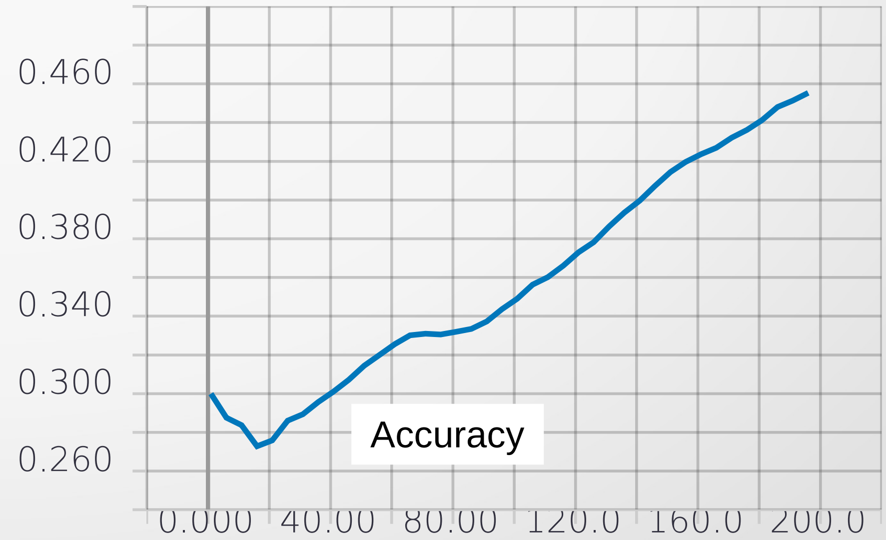
Baseline : CNN



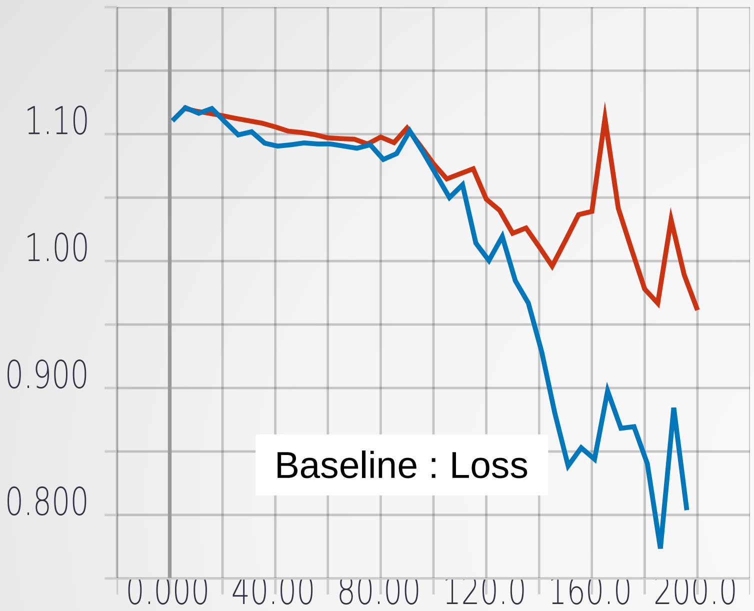


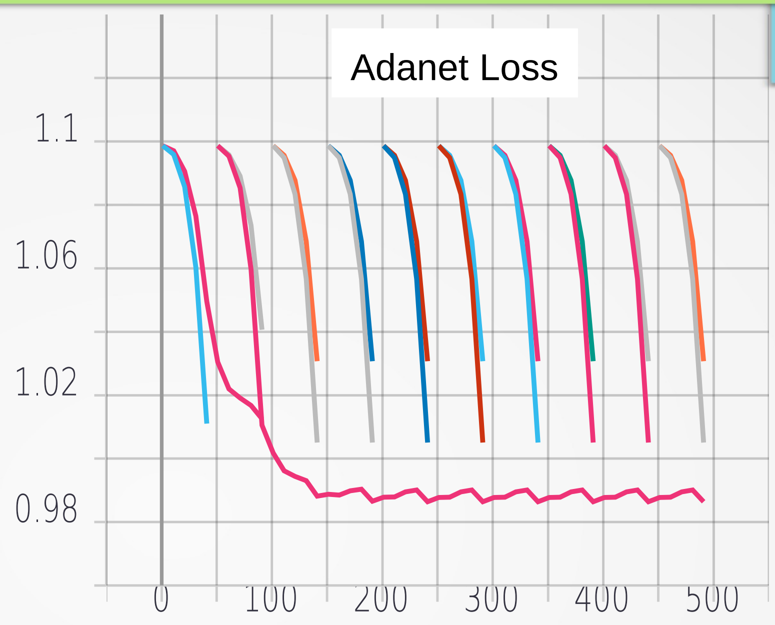
CNN network
Conv layers : 3
Dense layers : 1
Kernel : (5,5)
Filters : 32
Steps : 200

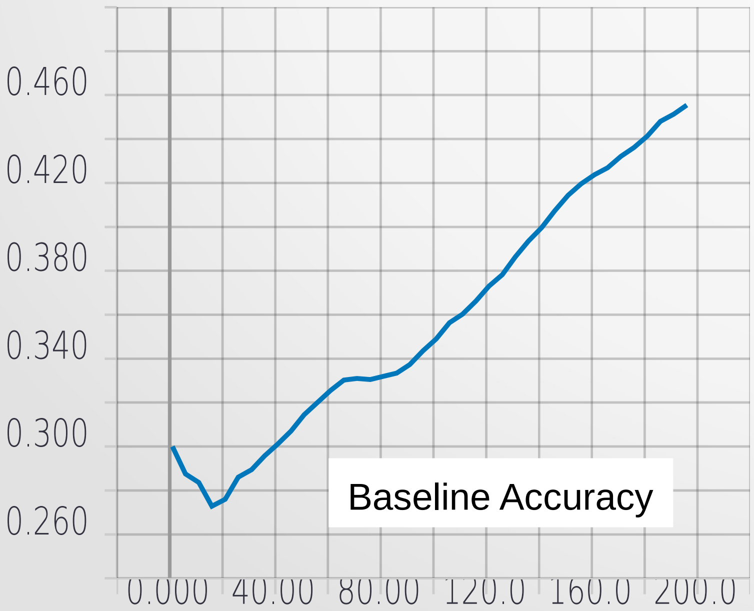
Train
Eval

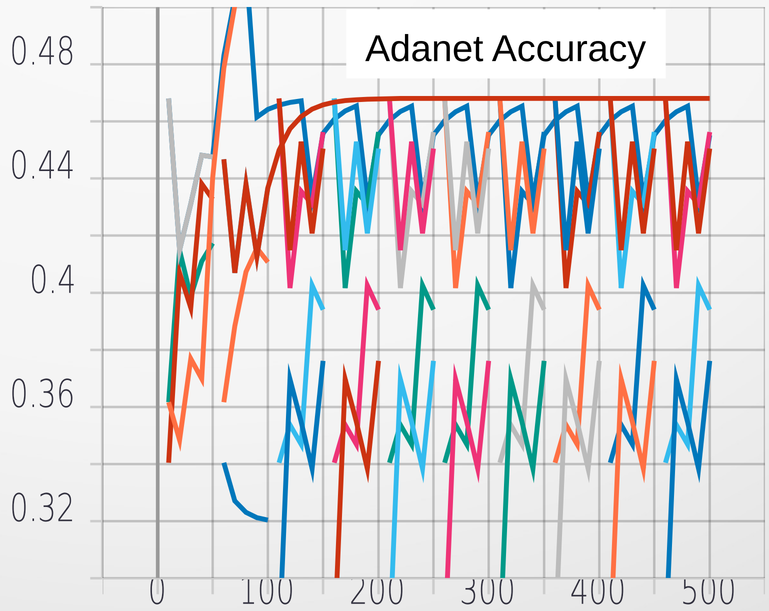


Adanet : CNN conv. layers

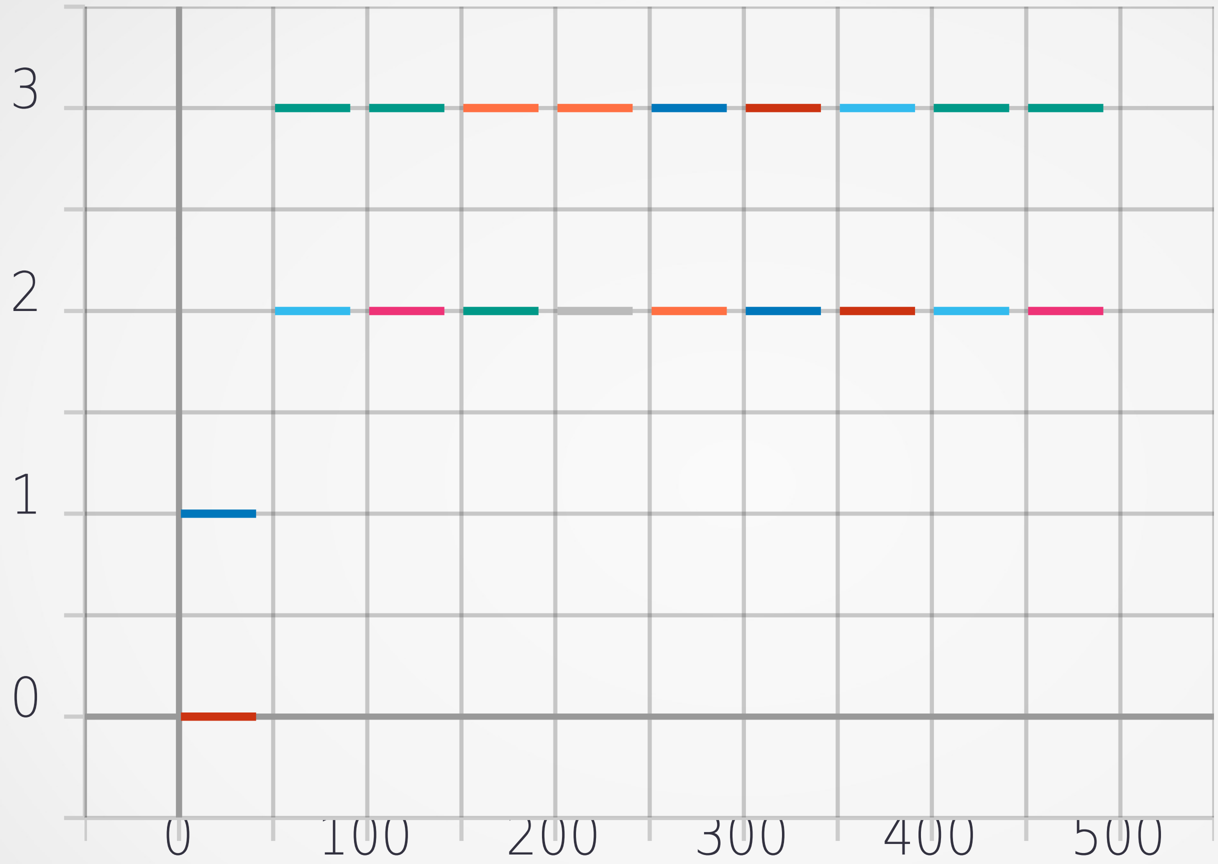






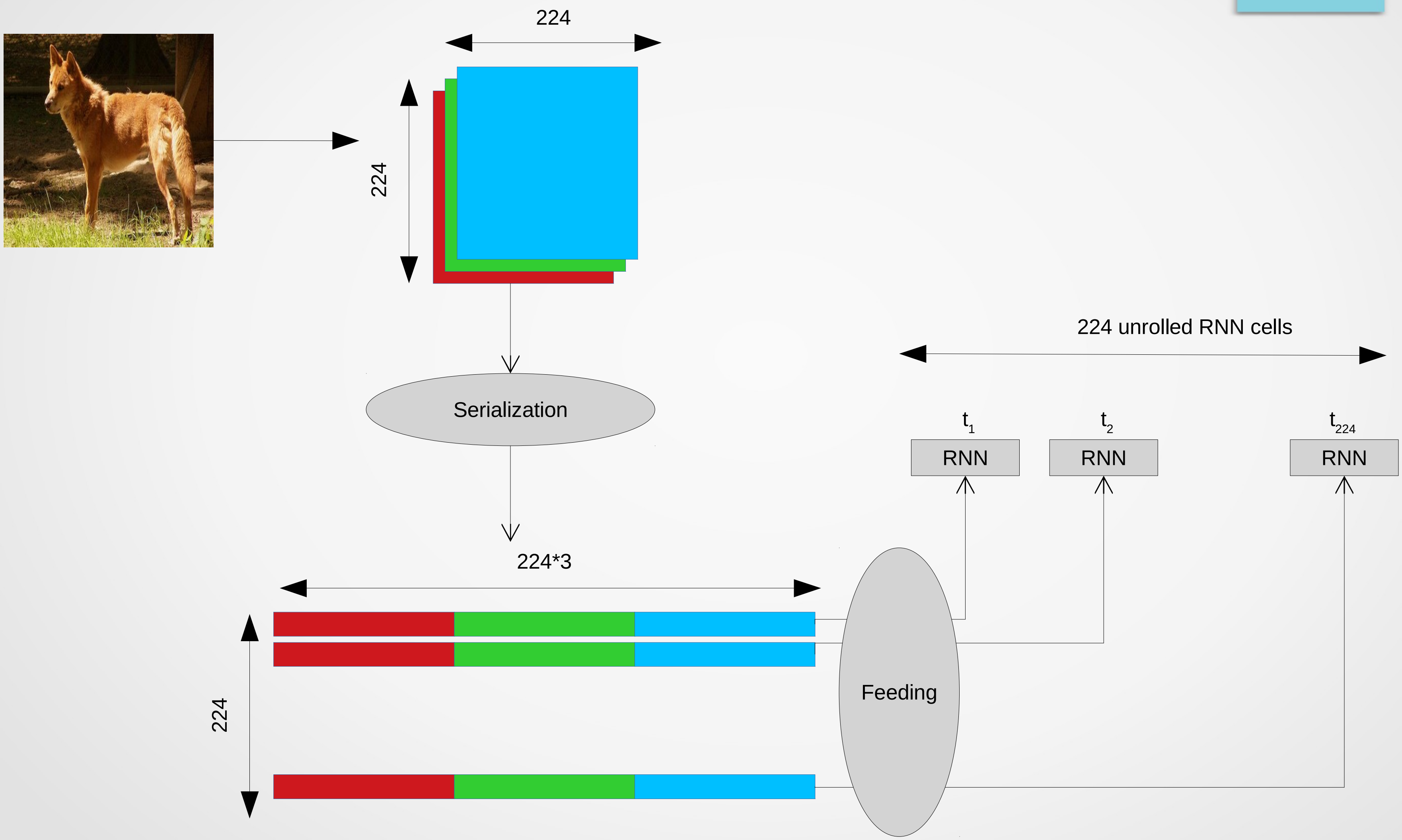


CNN: Adanet structure

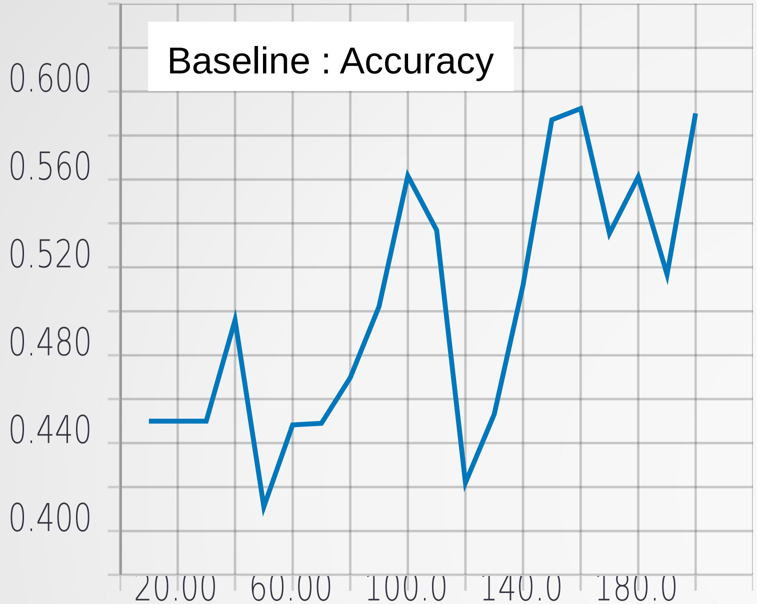


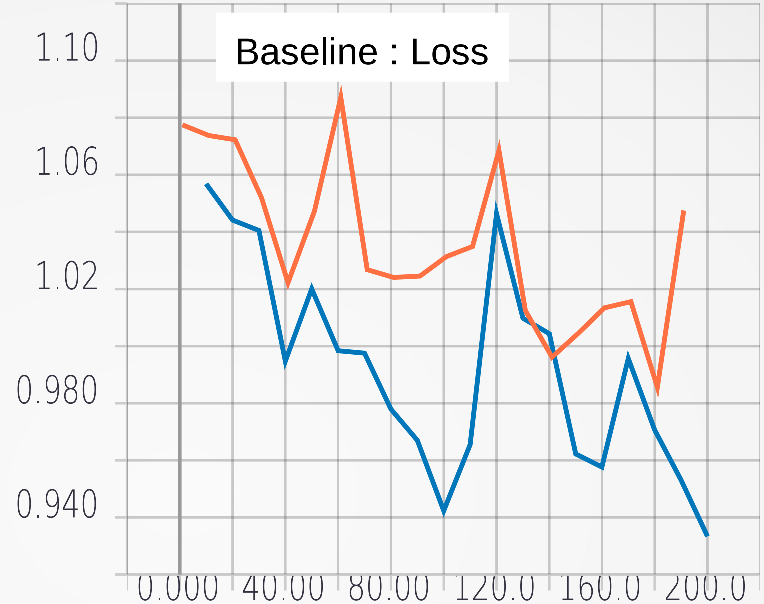
NN structure : convolutional layers
Rademacher regularization

RNN : Colored images serialization



RNN Baseline: Simple RNN cell

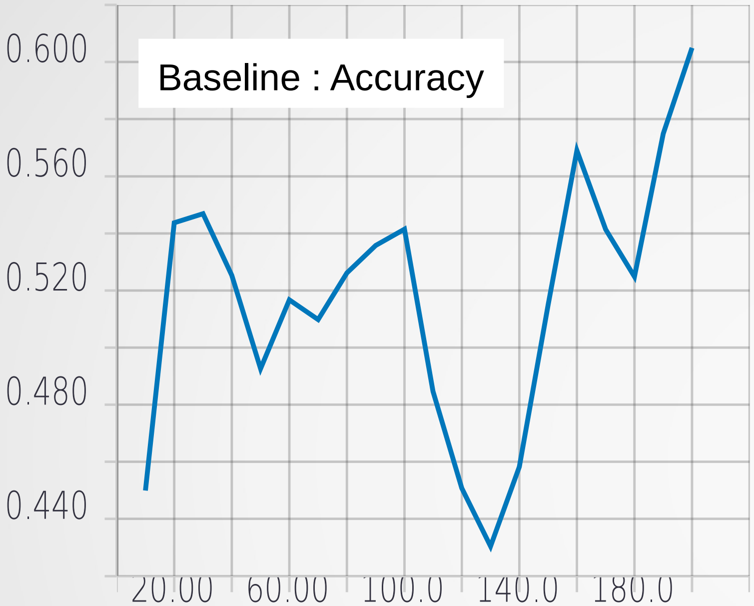


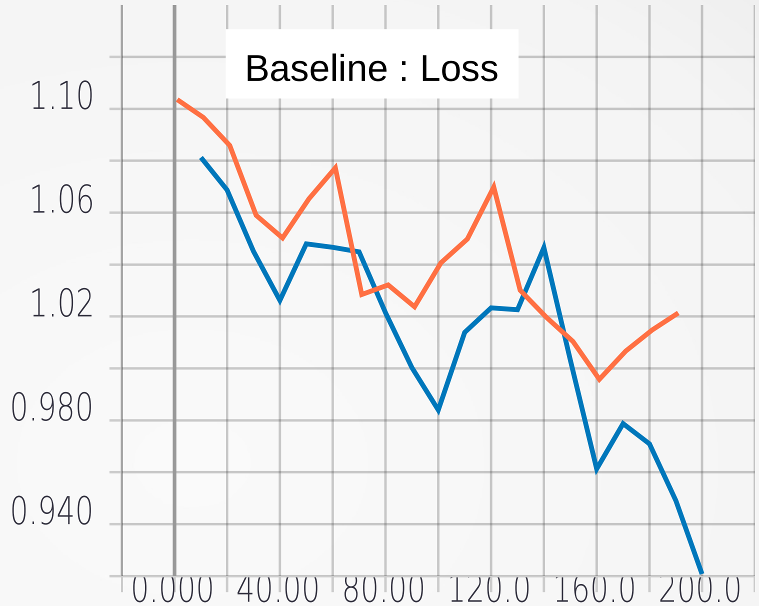



```
NN type                : ............................ RNN
Units in dense layer : ............................ 10
Number of layers       : ............................ 1
Dropout rate           : ............................ 0.0
Seed value             : ............................ 42
Nb of classes (logit): ............................ 3
Weights initializer  : ............................ truncated_normal
Batch normalization  : ............................ True


Cell type              : ............................ RNN
Hidden units           : ............................ 128
Stacked cells          : ............................ 1
Time steps             : ............................ 224
```

```
Time (sec) 86.21402406692505
RNN_EVAL_ACCURACY: 0.699999988079071
LOSS: 0.903887152671814
```
GLOBAL_STEP: 200

RNN Baseline : LSTM / 2 layers

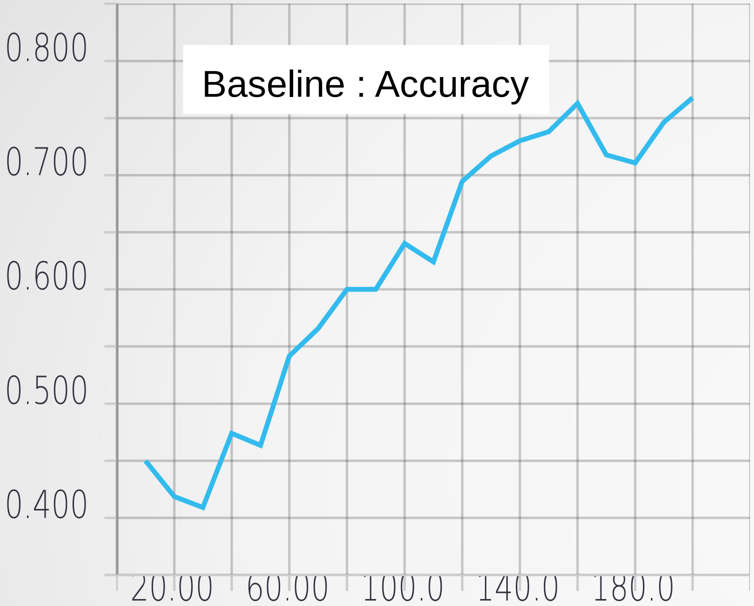


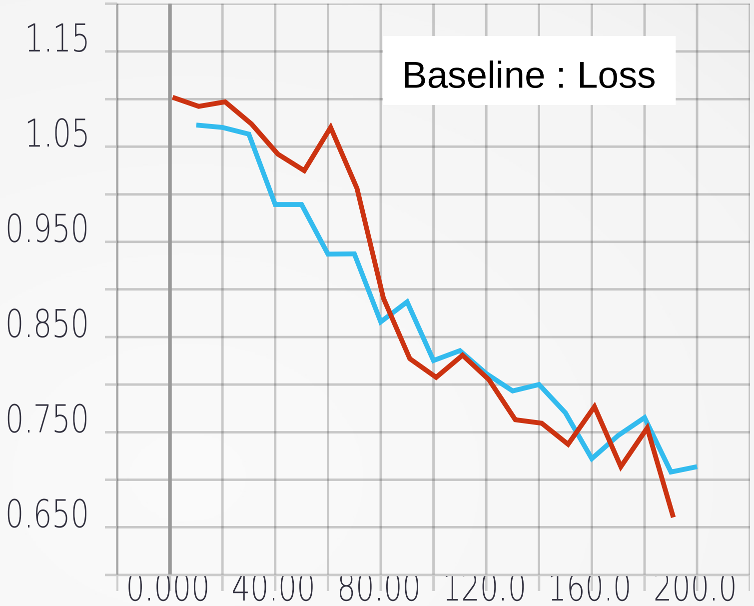



```
NN type              : ............................ RNN
Units in dense layer : ............................ 10
Number of layers     : ............................ 2
Dropout rate         : ............................ 0.0
Seed value           : ............................ 42
Nb of classes (logit): ............................ 3
Weights initializer  : ............................
truncated_normal
Batch normalization  : ............................ True

Cell type            : ............................ SLSTM
Hidden units         : ............................ 128
Stacked cells        : ............................ 2
Time steps           : ............................ 224
```

```
Time (sec) 565.1023576259613
RNN_EVAL_ACCURACY: 0.6499999761581421
LOSS: 0.8784809112548828
GLOBAL_STEP: 200
```

RNN Baseline : GRU 2 layers





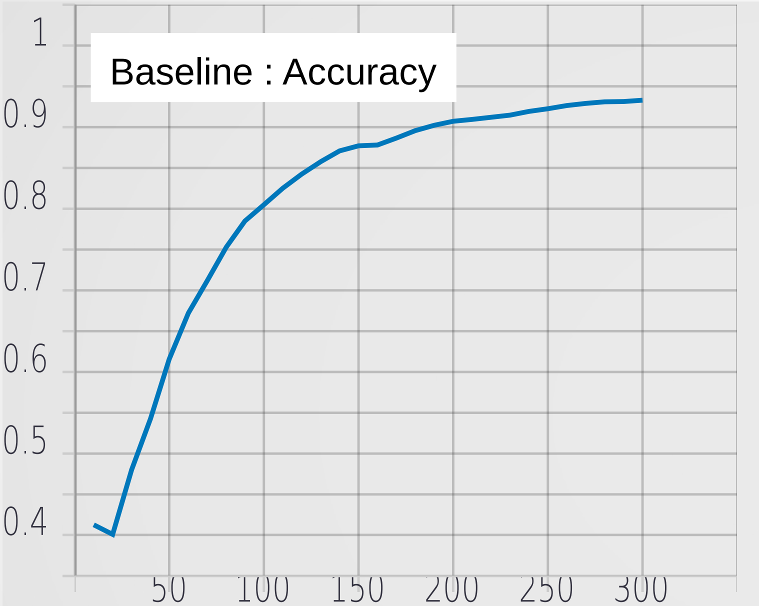

```
NN type               : ............................ RNN
Units in dense layer : ............................ 10
Number of layers      : ............................ 2
Dropout rate          : ............................ 0.0
Seed value            : ............................ 42
Nb of classes (logit): ............................ 3
Weights initializer   : ............................ truncated_normal
Batch normalization   : ............................ True

Cell type             : ............................ SGRU
Hidden units          : ............................ 128
Stacked cells         : ............................ 2
Time steps            : ............................ 224
```

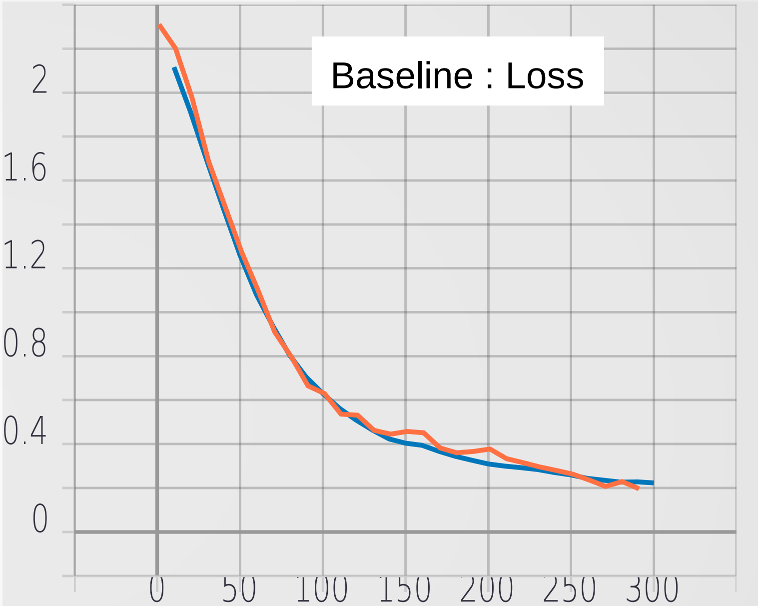
```
Time (sec) 1384.7687520980835
RNN_EVAL_ACCURACY:
0.800000011920929
LOSS: 0.7223628759384155
GLOBAL_STEP: 200
```

MNIST dataset

Train dataset : 55 000 inputs
Test dataset : 10 000 inputs
Classes : 10

Baseline : GRU & MNIST

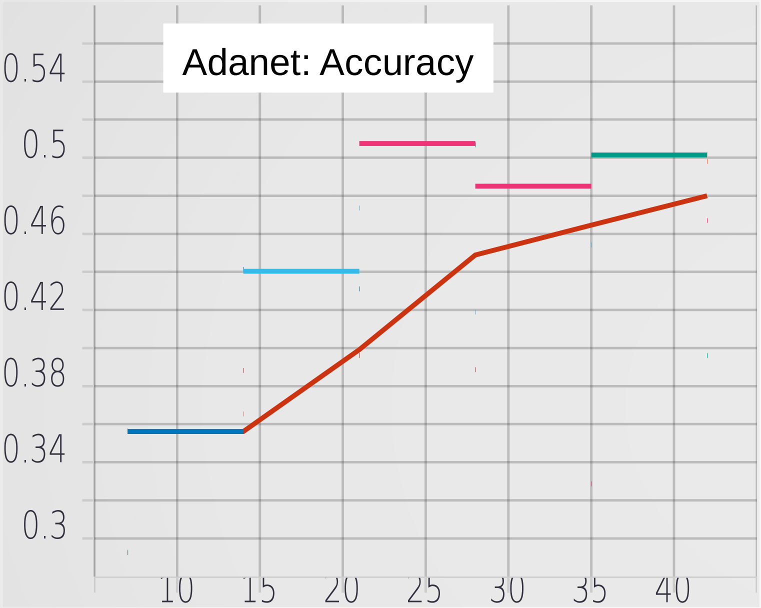


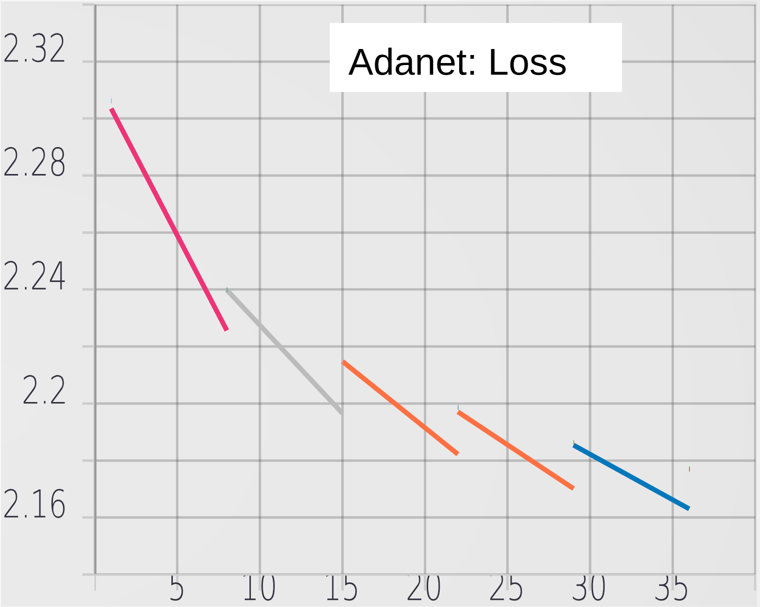



```
Global steps  : .................................... 300
NN type              : ............................ RNN
Features shape       : ............................ (28, 28)
Number of layers     : ............................ 2
Dropout rate         : ............................ 0.0
Seed value           : ............................ 42
Nb of classes (logit): ............................ 10
Weights initializer  : ............................ truncated_normal
Batch normalization  : ............................ True
Learn mixture weights: ............................ True
Cell type            : ............................ SGRU
Hidden units         : ............................ 128
Stacked cells        : ............................ 2
Time steps           : ............................ 28
```

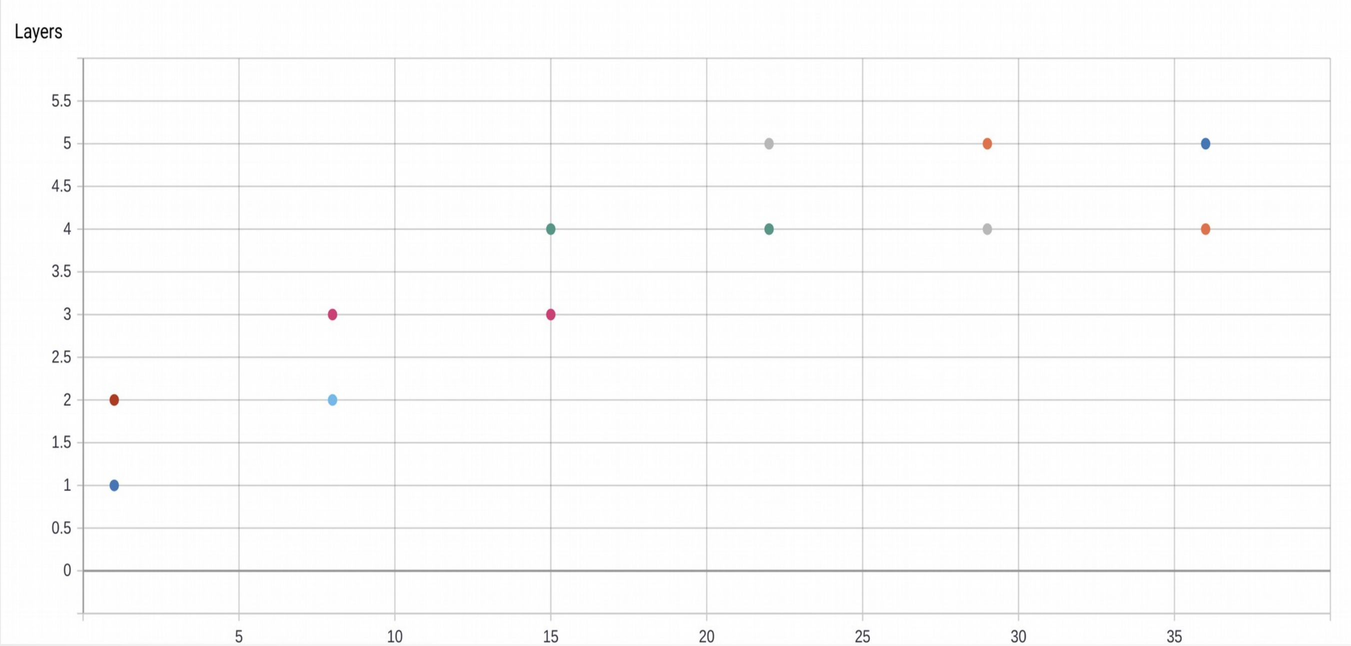
```
Time (sec) 167
RNN_EVAL_ACCURACY: 0.94
LOSS: 0.21498049795627594
GLOBAL_STEP: 300
```

ADANET : GRU & MNIST





ADANET : architecture




```
Global steps  : .................................... 300
NN type              : ........................... RNN
Features shape       : ........................... (28, 28)
Adanet boosting iter.: ........................... 40
Adanet iter per boost: ........................... 7
Dropout rate         : ........................... 0.0
Seed value           : ........................... 42
Nb of classes (logit): ........................... 10
Adanet regularization: ........................... 1e-05
Weights initializer  : ........................... truncated_normal
Batch normalization  : ........................... True
Learn mixture weights: ........................... True
Cell type            : ........................... SGRU
Hidden units         : ........................... 128
Stacked cells        : ........................... 2
Time steps           : ........................... 28
```

Layers: 5 stacked GRU cells

Classes : 10

Conclusions

- Formulation allows flexible ‘feed-forward’ architectures
- Formulation allows increase number of units in layers as well
- Adapts complexes NN models to complex problems
- No prior selection of model complexity (number of layers, depth of tree,…)
- Apply to any kind of ensemble models (provide layers or logits)

- Candidates fine tuning :
  - Dropout inside candidates subnetworks

- CNN
  - Strategy over convolutional layers : $C = (I-F+2P)/S +1$
  - Strategy for increasing dense layers

- RNN
  - Strategy to be tuned
  - Requires more resources

- GAN networks
  - To be investigated