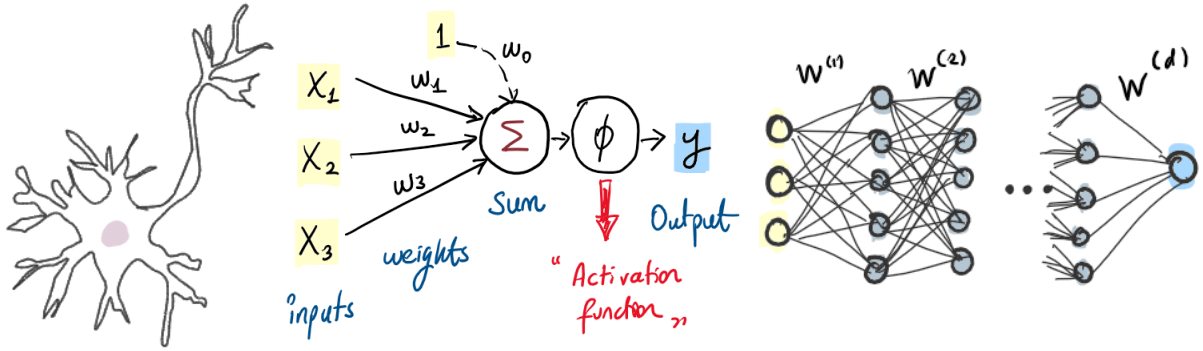
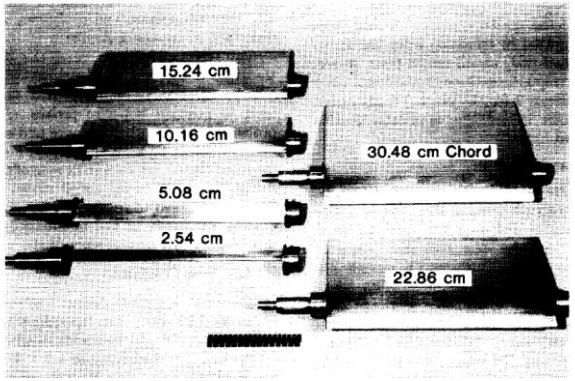
Data Driven Engineering I: Machine Learning for Dynamical Systems

Introduction to Deep Learning: Review

Institute of Thermal Turbomachinery
Prof. Dr.-Ing. Hans-Jörg Bauer



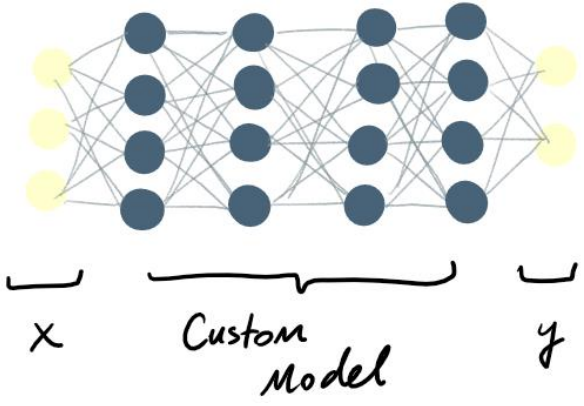


Neural Networks

* X → N.N → y

⇓

" Predict + Compare + Learn "

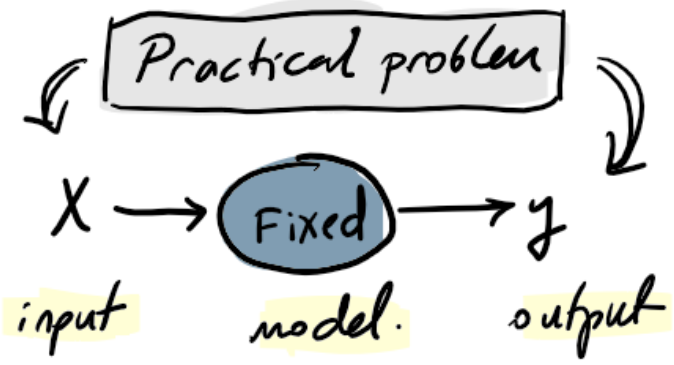


* Custom ⇨ You need to design it !
  ↳ " Versatile "

* How to think about NN ?
  ↓
  " Dynamical ensemble learner "

What we did so far ?

- "Non-linear" Difficult Problem



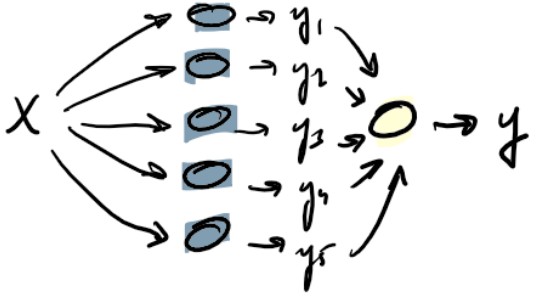
- "Week 2-5" ⇒ Model → Reg. / Class. / Cluster. / Dim. Red.

- $y = w \cdot x$ ⇒ ✗
- Error is too high!
- Simple methods cannot capture it;

⇒ I need a more complex model.

What you can do now?

- "Non-linear" Difficult Problem



fixed model

fixed set of rules (RF)

⇨ Ensemble learners

↳ A set of simple models can solve a more difficult problem.

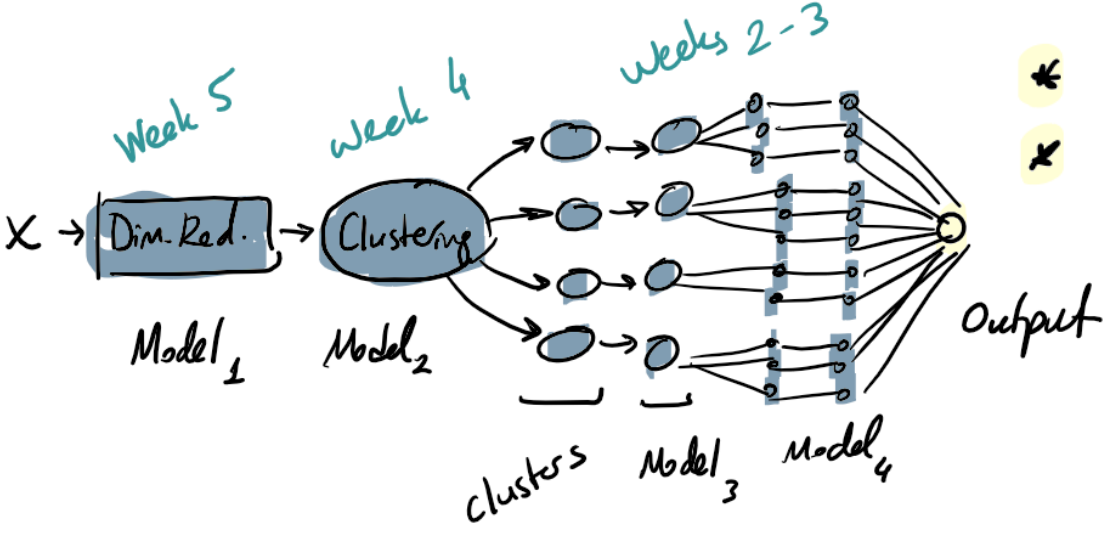
"Random Forests"

X

↳ I need a more complex model.

What you can do now?

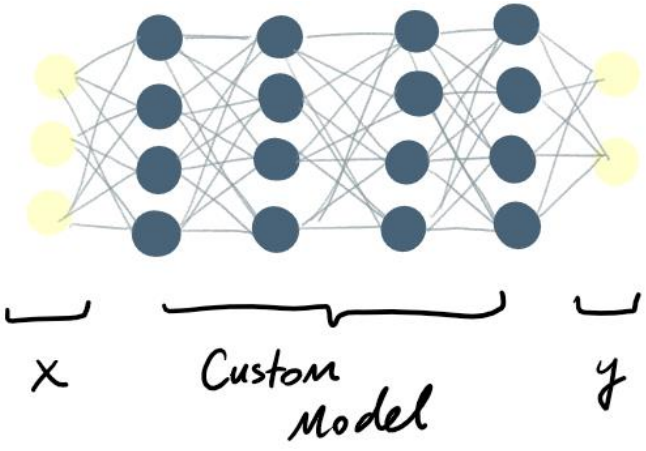
* Ensemble of shallow models



<< Project Phase I >>

* User-defined architecture
* User-defined connectivity

How to interpret NN as a model?



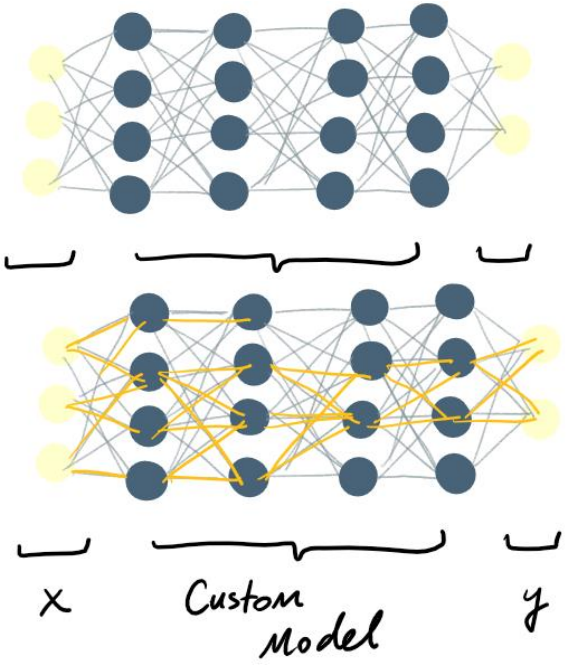
(i) I have a difficult problem.

(ii) My efforts to create a custom model failed;
↳ Error is high // cannot generalize well.

(iii) "What if I learn how to create an ensemble model from data?"

Learning Task → X → y
"Model"

How to interpret NN as a model?



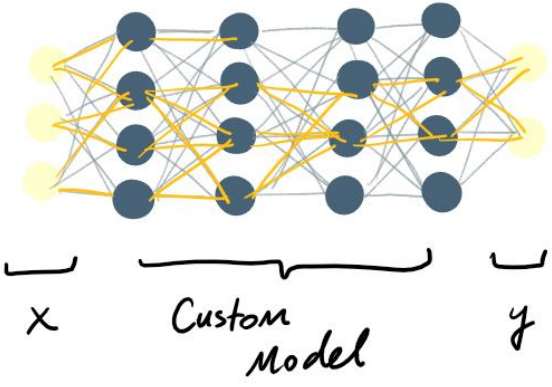
Solution:

(I) Create a probability space for candidate models

Learning Task → X → y
→ "Model"

(II) Learn how to activate "right pathways" for right combinations

How to interpret NN as a model?

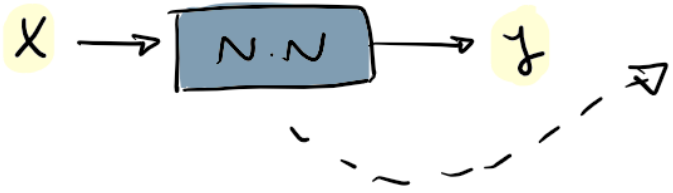


Give me a model that learns

- looking from the right perspective (DR),
- Group observations (clustering)
- Change # dimensions as needed (DR)

- Create a set of ensamble learners → hierarch. connections
  → dependent, complex.

How to create a custom model ?

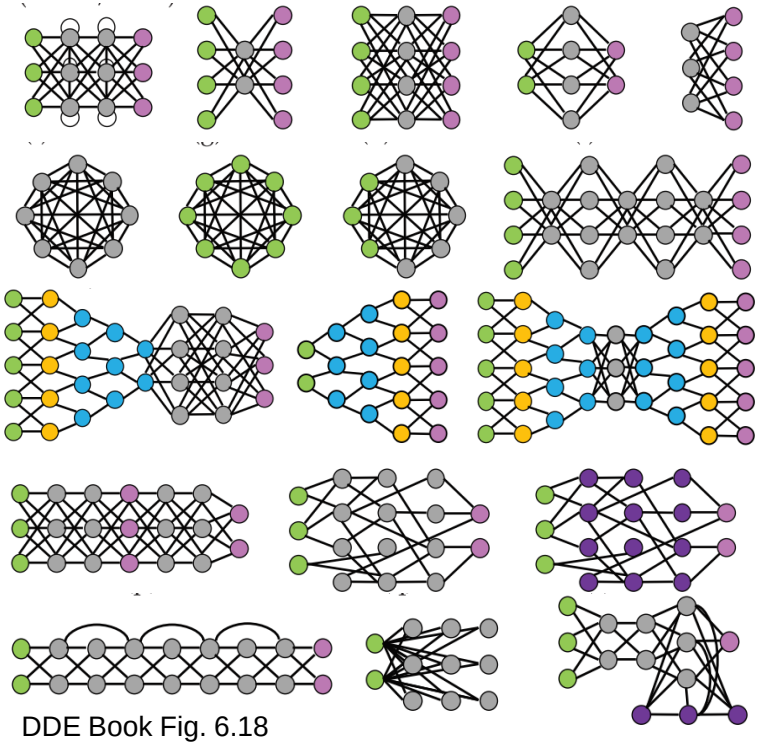


"Predict + Compare + Learn"

* Prediction Step ⇒ Architecture

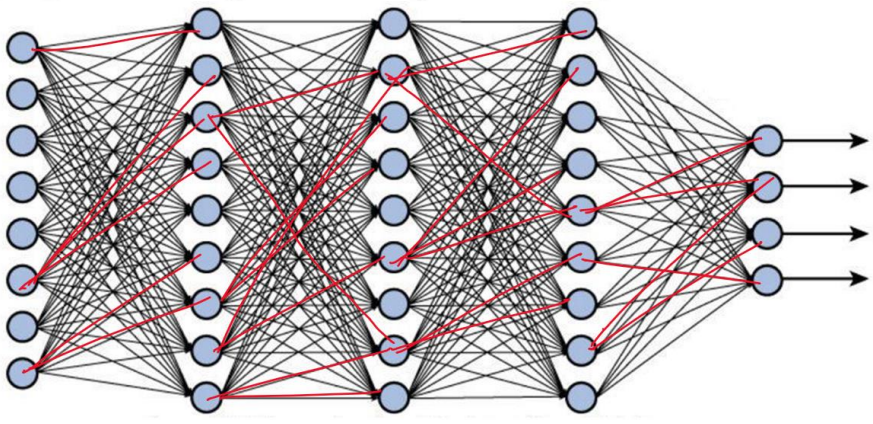
- # layers
- # nodes
- # Connections → "Dropout"
- Activation function → "Initialization"

Architecture Regulation:



DDE Book Fig. 6.18

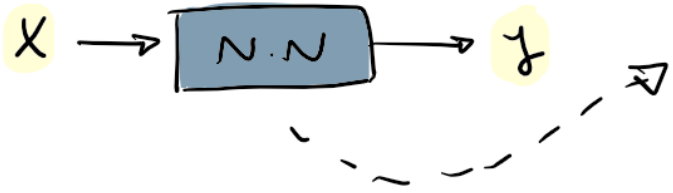
- # layers
- # neurons / layer
- Connections
- Dropouts (~20–50 %)



Ateliers & Saveurs in Montreal

colab

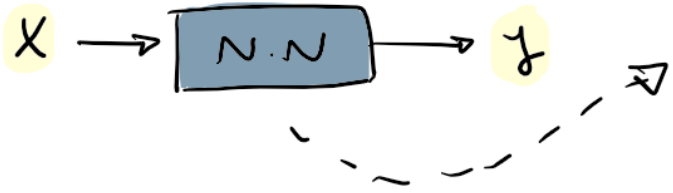
How to create a custom model ?



" Predict + Compare + Learn "

* Comparison Step ⇒ Error criteria

How to create a custom model ?



" Predict + Compare + Learn "

* Learning Step

- Batch / Epoch numbers
- Learning Rate
- Weight Regularization (penalty-based)
- Early stopping

Penalty-based Regularization

- The common approach in ML Algorithms
- $E_{total} = E_{data} + E_{Reg}$
- Obj: Force the weights take smaller values
- TF → Keras → l_1, l_2, l_1-l_2 Reg.

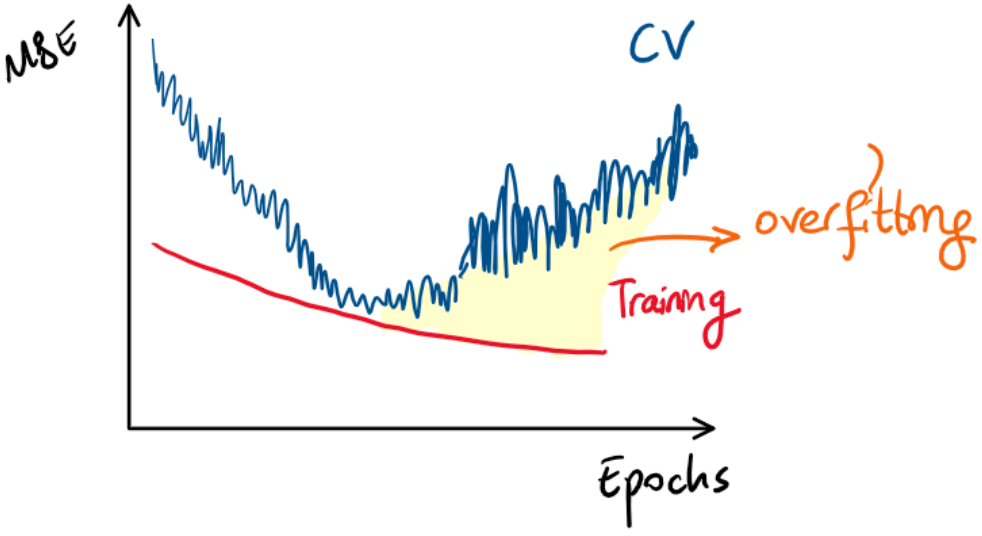
Example:

- $Loss = \sum (y_t - y_p)^2 + \lambda \sum w_i^2$ (λ: hyperparameter)
- $w_{t+1} = w_t + \delta \frac{\partial E}{\partial w_t}$
- $w_{t+1} = w_t (1 - \delta\lambda)$ Forgetting mechanism

Ateliers & Saveurs in Montreal

colab

Early Stopping:



* As training (epochs) continues, model would overfit the training data.

* A natural solution:
  - ✓ Keep track of best CV scores
  - ✓ Stop if there is no more improvement.

Tensor Flow:

* callbacks.ModelCheckpoint
* callbacks.EarlyStopping

Ateliers & Saveurs in Montreal

colab