A.L.S.E.
http://www.alse-fr.com

Formation Pratique
Practical Deep Learning

Présentation

La formation *Practical Deep Learning* (Apprentissage Approfondi) est destinée aux ingénieurs ou aux programmeurs qui souhaitent acquérir rapidement les compétences indispensables pour mettre en œuvre dans leurs projets le *Deep Learning* et l'*Inférence*, avec des exercices pratiques concrets.
Ces exercices représentent environ la moitié du temps de formation.
N'hésitez pas à consulter cette vidéo pour la présentation de cette formation.

Vous trouverez de nombreux modèles commerciaux de réseaux neuronaux profonds « prêts à l'emploi ». Mais ce qu'on ne vous dit pas toujours est que l'effort et l'expertise requis pour collecter et optimiser un jeu de données d'apprentissage standard pour votre application est très voisin de celui requis pour créer ce modèle en partant de rien !

Cette formation de 5 jours vous donnera toutes les bases et la pratique pour démarrer vos projets sans délai et réduire la durée de votre apprentissage en jours plutôt qu'en mois.

Ce cours est basé sur le langage **Python** et utilise intensément l'API de réseaux neuronaux **Keras**, qui est l'API recommandée pour l'environnement **TensorFlow**.
Les librairies Numpy, Matplotlib, Pandas, Scikit-learn, and TensorBoard sont également largement utiliseés.
Bien que basée sur Keras / Tensorflow, les principes et concepts enseignés seraient applicables à toute autre librairie ou environnement de *Deep Learning*.

Notez que nous proposons aussi une formation **Python** de trois jours (dispensée en Français) que vous pouvez souhaitez suivre avant la formation *Deep Learning*.

Les **Instructeurs** sont aussi et surtout des **Experts reconnus** qui utilisent pratiquement les Langages et les Méthodes de Conception qu'ils enseignent. Ils savent partager leur savoir-faire avec passion et sont particulièrement appréciés des participants.

Pré-requis

Cette formation demande des **connaissances préalables importantes** dont :

- ✔ La maîtrise d'au moins un langage de programmation de haut niveau orienté Objet (Python, Ruby, C++, C#, Java etc). La maîtrise de Python est un plus mais n'est pas rigoureusement indispensable.
- ✔ La compréhension de nombreux concepts mathématiques : fonctions continues multivariables, fonctions linéaires et non-linéaires, exponentielles, les bases du calcul différentiel, les bases de statistiques -moyenne, écart-type, variance, probablilités, distributions, histogrammes-, les bases de l'algèbre linéaire matrices sommes et produits...

Malgré tout, cette formation ne demande pas de forte compétences sur ces sujets, il n'y aura pas à écrire ni résoudre d'équations ni à établir des démonstrations ! Il s'agira de mettre en œuvre ces concepts mathématiques à travers de la programmation.

Durée

Cinq (5) jours soit **35 heures effectives** de cours

Lieu

Dans les locaux d'ALSE, **Paris XIII**ème

NB : à partir de quatre ou cinq stagiaires, une formation sur site est envisageable.

ALSE est un Organisme de Formation Professionnelle Continue déclaré auprès de la DIRECCTE depuis 1996 sous le numéro 26.21.01281.21 et enregistrée dans le DataDock.
Cette formation peut donc être prise en charge dans ce cadre.

Objectifs pédagogiques

- Mettre en œuvre les bases de la programmation Python en utilisant le Jupyter Notebook
- Les principes et pratiques de l'apprentissage supervisé et de l'apprentissage profond
- Comment utiliser les réseaux neuronaux pour les problèmes de classification et de régression.
- Utiliser l'apprentissage non supervisé pour réduire la visualisation et la dimensionnalité.
- Utiliser les réseaux neuronaux convolutionnels pour les classifications d'images.
- Utiliser Keras et TensorBoard
- Utiliser l'inférence au travers de réseaux neuronaux pré-construits
- Tirer avantage des réseaux neuronaux pré-entraînés par le transfert d'apprentissage
- Préparer and nettoyer les jeux de données pour le Deep Learning.
- Les concepts de l'Apprentissage Profond (Deep learning) et les techniques classiques comme les algorithmes de descente de gradient, courbes d'apprentissage, régularisation, dropout, normalisation de lot, architecture de création, réseaux résiduels, taille et quantification, l'architecture MobileNet, plongement lexical, et réseaux neuronaux récurrents.
- Introduction aux réseaux adverses génératifs et à la détection d'objets.
- Les principes qui sous-tendent l'offre croissante de flots spécifiques à différents vendeurs qui permettent de déployer des modèles de réseaux neuronaux pour l'inférence dans le cloud et dans les composants embarqués (« Edge »).

Programme

Journée 1

Introduction to Deep Learning

• AI versus ML versus Deep Learning
• "Classical" Machine Learning
• Deep Learning
• Supervised Learning • Unsupervised Learning
• Neural Networks
• Cloud versus Edge Computing
• Applications
• Libraries/Frameworks for Training
• Cloud Platforms for Training and Inference
• Vendor Platforms for Deploying ML / DL

Jupyter Notebook

• AWS Deep Learning AMI • Connect to Remote Machine using SSH
• Using Jupyter Notebook • Basic Markdown
• Output and Evaluating Expressions
• Expanding, Collapsing, Hiding Output
• Menus and Tool Bar

Python Basics

• Functions, Variables, and Values • Control Statements
• Operators • Imports • Instance Objects
• Kwd arguments • Range • Format • Tuples and Lists
• Functions returning Functions
• Numpy Array • Multi-dimensional Numpy array • Reshape • Slices
• One-Hot Encoding • Reduction Operations
• matplotlib.pyplot • List Operations

Linear Regression

• Regression Task • Defining a Model
• Cost / Loss / Error Function • Cost as a Function of Trainable Parameters
• Mathematical Optimization • Contour Plot of Cost Function
• Gradient Descent • Stochastic Gradient Descent

Keras

• TensorFlow and Keras • Versions of Keras
• A Keras Sequential Model
• Datasets
• Gradient Descent in Keras
• History Object • Plotting Progress

Logistic Regression

• Classification Task • One-Hot Labels
• The Hypothesis or Model
• Calculating the Cost Function • Converting Scores to Probabilities
• The Softmax Function
• Compare using Cross-Entropy • Multinomial Logistic Regression
• Plotting the Decision Boundary

Journée 2

Neural Networks

• Neural Networks
• An Artificial Neuron • Common Activation Functions
• A Deep Neural Network
• Forward and Back-Propagation
• Kinds of Neural Network

Non-linear Regression

• Linear Regression
• A Non-Linear Polynomial Model
• The Rectified Linear Unit (ReLU) • Normalizing the Data
• Exploding and Vanishing Gradients • Varying the Weight Distribution
• Xavier Glorot Initialization
• Non-Linear Keras Model
• The Magic of Deep Neural Networks

Non-linear Classification

• A Non-Linear Decision Boundary • Decision Boundary and Softmax
• Non-Linear Neural Network for Classification
• From ReLU to Decision Boundary
• Softmax

Overfitting and Regularization

• Training versus Test Datasets • Scikit-learn • Learning Curves
• Matching the Network to the Problem
• How to Reduce Overfitting? • More Data?
• Regularization (L2 Regularization) • Choosing Lambda • L2 versus L1 Regularization

Stochastic Gradient Descent

• Full-Batch vs Stochastic Gradient Descent • Mini-Batches
• The Landscape of the Cost Function • Stationary Points
• Learning Rate • Learning Rate Decay Schedule
• Momentum • Nesterov Momentum
• Adaptive Per-Parameter Learning Rates

• Adam Algorithm

Journée 3

Splitting the Dataset

• The MNIST Dataset
• A Deep Neural Network for Classification • Hyperparameters
• Training, Validation, and Test Datasets • K-Fold Cross-Validation • Validatation
• Choose a Single Scalar Metric
• Imbalanced Classes or Rare Events • ROC Curve
• Trading off Precision and Recall

Convolutional Neural Networks

• Convolution • Patch Size and Stride • Valid Padding versus Same Padding
• Network Size • Multiple Features Maps • Pooling
• Stride Versus Receptive Field • Hierarchical Feature Detection
• Filter Visualization • Number of Parameters and Values
• Plotting Convolution Filters

Visualization using TensorBoard

• Visualizing a Graph in TensorBoard • Visualizing Scalars in TensorBoard
• Visualizing Multiple Runs • Visualizing Weights and Activations
• Highlighting a Histogram
• Visualizing an Embedding in 3-D Space
• Using TensorBoard from Keras

PCA, t-SNE, and K-Means

• Dimensionality Reduction
• Principal Component Analysis • PCA for Visualization
• t-SNE for Visualization • Clustering with K-Means
• PCA for Dimensionality Reduction • Linear Regression Out-of-the-Box
• Normalizing the Dataset

Journée 4

Data Preparation

• The Deep Learning Process • Feature Engineering
• Correlated Features, Missing or Bad Values • Choose Meaningful Features • Avoid Magic Values
• Pandas Dataframe • Pandas Summary Statistics
• Pandas Scatter Matrix • Cleaning Data with Pandas
• TensorFlow Extended (TFX)
• Error Analysis • Artificial Data Synthesis
• Data Augmentation

Dropout and Batch Normalization

• Dropout • When to use Dropout?
• Batch Normalization • Scale-and-Shift
• Benefits of Batch Normalization
• Calculating the Scaling Factors

Inception and Residual Networks

• General Principles of Network Architecture • Evolution of CNN Architectures
• Principles of the Inception Architecture • Fully-Connected versus Sparse
• Inception Module
• Global Average Pooling • Fully Convolutional Network
• Residual Networks • Matching Dimensions
• Performance of Inception and ResNet

Transfer Learning

• Why Transfer Learning?
• Re-use trained weights • Simple Transfer Learning in Keras
• A Pre-trained Inception Network • Fine-Tuning Previous Layers
• Saving and Loading
• Make Each Class Visually Distinct • Other Tips for Image Data

Journée 5

Pruning and Quantization

• Inference Engines at the Edge
• ML / DL Tool Flow for Edge Computing
• Neural Network Exchange Formats
• Network Pruning • Quantization • 8-Bit Quantization
• TensorFlow Lite Post-Training Quantization
• OpenCV • OpenVX • OpenCL

MobileNet

• Introduction to MobileNet
• Depthwise-Separable Convolution
• MobileNet V1 Architecture
• MobileNet V2 Architecture
• Hyperparameters
• MobileNet Family • MobileNet Inference in TensorFlow
• CNNs Compared

Encoding and Word Embedding

• Coding Categorical Data • Binning
• Text in a Neural Network • Word Embedding and Semantics
• Hidden Layers and Latent Features
• The Word2vec Algorithm
• Negative Sampling Neural Network

Recurrent Neural Networks

• Recurrent Neural Network (RNN) • RNN Applications • Long Short Term Memory – LSTM • LSTM Gates • LSTM Connections • Gated Recurrent Unit – GRU • Simple Character-Level RNN in Keras • LSTM trained on Linux Source Code • Bidirectional LSTM

GANs

• Putting Networks End-to-End
• Generative Adversarial Network (GAN)
• Deep Convolutional GAN Generator
• Generated Images
• GAN to Generate MNIST Digits

Object Detection

• Datasets for Object Detection
• Networks for Object Detection
• Object Detection – Faster R-CNN • Training a Faster R-CNN Network
• Image Segmentation using Mask R-CNN
• Pixel-Level Image Segmentation