ETH Zürich
Prof. Dr. Josef Teichmann

February 9, 2023

Exam for the lecture

"Machine Learning in Finance"

401-3932-19L

Please fill in the following details:

Surname		**Given name**		**Legi-number**								**Prüfungsnr.**		
				X	X									

the first two letters for each box — *last six numbers* — *Do not fill this out*

Important:

- Fill in the first two letters of your surname, your given name and the last six numbers of your legi-number
- Put your student card on the table
- The duration of the exam is **90 minutes**. Before these 90 minutes start, you are allowed to read the exam for **10 minutes** during which no writing is allowed. Follow the commands indicating whether you can start reading and start writing. When you are told to **stop** writing, put down your pens immediately.
- Begin each exercise on a new sheet of paper, and write your name on each sheet. There are 4 questions in total.
- Only pen (ink) and paper are allowed. The ink must not be red or green. Do not use whiteout, instead just cross out the relevant parts.

Please do <u>not</u> fill in the following table

Question	**Points**	**Control**	**Maximum**
#1			
#2			
#3			
#4			
Total			

Good luck!!! **Please turn the page!**

Some hints:

1. Recall that we denote with $\mathcal{C}(\mathcal{X})$ the set of all continuous functions $f : \mathcal{X} \to \mathbb{R}$. If $\mathcal{X} = \{x_1, \dots, x_n\}$ is some finite set, then we can identify $\mathcal{C}(\mathcal{X})$ with $\mathbb{R}^n$.

2. In Question 2-4, you will have to write pseudo code. This means that you are supposed to describe in a structured way how your algorithm works. It is not necessary to have precise syntax. Hence, you can also use mathematical notation if it is clear how it is meant (e.g. subscripts, powers and standard functions like $\sqrt{\cdot}$ etc). One example would be that if you have a vector $\alpha \in \mathbb{R}^n$, you write the sum of the elements either by (you can use Greek letters etc)

   ```
   value_sum = $\sum_{i=1}^{n} \alpha_i$
   ```

   or you could write

   ```
   value_sum = 0
   for i=1,...,len(α):
           value_sum = value_sum + α_i
   ```

   where these are just examples. Another example of a function you may use is the function `cumsum` that computes the cumulative sum of a vector $v \in \mathbb{R}^k$, i.e.

   ```
   cumsum(v) = w ∈ R^k with w_i = \sum_{j=1}^{i} v_j .
   ```

   You may further use any `python` (including `numpy`), `Matlab` or `R` syntax, i.e. in order to define a vector $\tau = (0, 1, \dots, n)$, you can e.g. write either of the following without further commenting,

   - `τ = linspace(0,n,n+1)`
   - `τ = arange(n+1)`
   - $\tau = (0, 1, \dots, n)$

   Use a % to write comments, e.g. if you explain what a function you use does. You can structure your code by defining functions, i.e. the following is admissible:

   ```
   % randomnormal(k) returns a random vector of
   % length k that is standard normal distributed
   v = randomnormal(10)
   solution1, solution2 = foo(v)

   function foo(x):
           % sin and cos for a vector x is applied
           % component wise
           return (sin(x), cos(x) )
   ```

   which generates a standard normal random vector $v \in \mathbb{R}^{10}$, and returns two vectors with components being the sin (stored in solution1) and cos (stored in solution2) of the corresponding components of the random vector. The goal of your pseudo code is only to show how your algorithm works, so do not spend time on thinking about whether this syntax is correct in some programming language.

Question 1. (10 points)

(a) (3 points) Universal Approximation Theorem (UAT):

i. Let $C(K,\mathbb{R})$ denote the continuous functions on the unit cube in $\mathbb{R}^d$. Let A be a subset of $C(K,\mathbb{R})$. Under which conditions can we conclude that A is dense?

A. A is a vector subspace additionally closed under multiplication and containing the constant function 1.

B. A is a vector subspace which additionally separates points.

C. A is a vector subspace additionally closed under multiplication, containing the constant function 1, which separates points.

D. A is closed under multiplication, containing the constant function 1, which separates points.

ii. Let $\mathcal{NN}$ be the set of shallow neural networks in $C(K,\mathbb{R})$ with RELU activation.

A. $\mathcal{NN}$ is a vector space but not separating points.

B. $\mathcal{NN}$ is a vector space closed under multiplication.

C. $\mathcal{NN}$ is a vector space additionally separating points.

D. $\mathcal{NN}$ is not a vector space but separates points.

iii. Let $\mathcal{NN}$ be the set of shallow neural networks in $C(\mathbb{R},\mathbb{R})$ with respect to RELU activation. Let $\mathcal{L}$ be the set of shallow neural networks on the unit cube K in $\mathbb{R}^d$ with respect to RELU activation. Denote by $\mathcal{NN}(\mathcal{L})$ the vector space of linear combinations all compositions $f \circ l$ of a network $f \in \mathcal{NN}$ with a function $l \in \mathcal{L}$, i.e. neural networks with two hidden layers.

A. $\mathcal{NN}(\mathcal{L})$ is dense in $C(K,\mathbb{R})$.

B. $\mathcal{NN}(\mathcal{L})$ is only dense in $C(K,\mathbb{R})$ if it is additionally closed under multiplication.

C. $\mathcal{NN}(\mathcal{L})$ is only dense in $C(K,\mathbb{R})$ if it is additionally closed under multiplication and point separating.

D. $\mathcal{NN}(\mathcal{L})$ is not dense in $C(K,\mathbb{R})$.

(b) (3 points) Signatures:

i. Let $u : [0,T] \to \mathbb{R}^{d+1}$ be a bounded variation curve, i.e. integrals along the curve are well defined. Component 0 equals the square of time itself, i.e. $u^0(t) = t^2$, and the curve starts at 0. By Signature Sig we mean the vector of all iterated integrals integrated up to time T.

A. Sig determines the curve $(u^1,\dots,u^d)$ uniquely.

B. Sig determines the curve $(u^1,\dots,u^d)$ up tree like equivalences.

C. Sig determines only the end point of the curve $(u^1,\dots,u^d)$.

D. Sig determines the curve $(u^1,\dots,u^d)$ up to a constant factor.

ii. Consider on path space of bounded variation curves on $[0,T]$ starting at 0 with zeroth component equal time itself all possible linear combinations of signature components. We denote this set by A.

A. A is a point separating vector space which is closed under multiplication.

B. A is a point separating vector space but not closed under multiplication.

C. A is a vector space closed under multiplication but not point separating.

D. A is a vector space but neither point separating nor closed under multiplication.

iii. What is the number of signature components up to depth M for a curve $u : [0, T] \to \mathbb{R}^d$ (M fold iterated integrals).

A. $\frac{(d)^{M+1}-1}{d-1}$

B. d^{M+1}

C. d^M

D. M^d

(c) (4 points) Definitions: Give the precise definition of a discriminatory and a sigmoidal (activation) function $\sigma : \mathbb{R} \to \mathbb{R}$. Define shallow and deep neural networks and provide a formula for the number of free parameters for a deep network with two hidden layers from $[0, 1]^{200}$ to $\mathbb{R}^{10}$.

Question 2. (10 points)

(a) (3 points) Financial Markets:

i. Let $S_t = (S_t^0, \dots, S_t^d)$ denote a random vector of asset prices at time t adapted to the information filtration $\mathcal{F}_t$, for $t = 0, \dots, N$ and let φ_t denote a vector of holdings in each asset (adapted to $\mathcal{F}_{t-1}$). We denote by $V_t = \sum_i \varphi_t^i S_t^i$ the value of the portfolio at time t. When is this portfolio self-financing?

A. It is never self-financing, since we did not specify a bank account.
B. If $\sum_i \varphi_{t+1}^i S_t^i = \sum_i \varphi_t^i S_t^i$ for $t = 0, \dots, N-1$.
C. If $\sum_i \varphi_t^i S_{t+1}^i = \sum_i \varphi_t^i S_t^i$ for $t = 0, \dots, N-1$.
D. It is always self-financing.

ii. Let φ be a self-financing portfolio in a financial market with bank account $S^0 = 1$ (zero interest rate) with value process V:

A. $V_{t+1} - V_t = \sum_i \varphi_{t+1}^i (S_{t+1}^i - S_t^i)$ for $t = 0, \dots, N-1$.
B. $V_{t+1} - V_t = \sum_i \varphi_t^i (S_{t+1}^i - S_t^i)$ for $t = 0, \dots, N-1$.
C. $V_{t+1} - V_t = 0$ due to the self-financing condition for $t = 0, \dots, N-1$.
D. The expression $V_{t+1} - V_t = \sum_i \varphi_{t+1}^i S_{t+1}^i - \varphi_t^i S_t^i)$ cannot be simplified.

iii. Let S be a financial market with a general bank account process S^0:

A. The market is free of arbitrage if there is no self-financing portfolio with $V_0 = 0$ and $V_N \geq 0$ and $V_N \neq 0$.
B. The market is free of arbitrage if for one self financing portfolio there is an equivalent martingale measure.
C. The market is free of arbitrage if there is a self-financing portfolio, which looses, i.e. $V_0 = 0$ and $V_N \leq 0$ and $V_N \neq 0$.
D. Absence of arbitrage can only be characterized by the existence of martingale measures for discounted markets and not by portfolio value processes.

(b) (3 points) Deep Trading:

i. Let S be a financial market with general bank account $S^0 = 1$ on a finite probability space, where the filtration is generated by the price process itself. How can we represent *any* self-financing trading strategies φ by neural networks:

A. We write φ_t^i as a neural network of S_{t-1} for $t = 1, \dots, N$ and $i = 1, \dots, d$.
B. We write φ_t^i as a neural network of S_{t-1} for $t = 1, \dots, N$ and $i = 0, \dots, d$.
C. We write φ_t^i as a neural network of $S_{t-1}, \dots, S_0$ for $t = 1, \dots, N$ and $i = 0, \dots, d$.
D. We write φ_t^i as a neural network of $S_{t-1}, \dots, S_0$ for $t = 1, \dots, N$ and $i = 1, \dots, d$.

ii. Let $u : \mathrm{dom}(u) \to \mathbb{R}$ be a utility function. What does this precisely mean?

A. u is monotone and concave.
B. u is monotone.
C. u is concave.
D. u is differentiable with positive first derivative.

iii. Let S be a financial market with bank account $S^0 = 1$ (zero interest rate) on a finite probability space, where the filtration is generated by the price process itself. Consider now a utility function $u : \mathbb{R}_{>0} \to \mathbb{R}$ and denote by

V_N the value of a self-financing portfolio at time N with initial capital x. What is the objective for the expected utility optimization problem?

A. $u(V_N)$.

B. $u(V_N + x)$.

C. $E[u(V_N)]$.

D. We have to choose an equivalent martingale measure Q and calculate $E_Q[u(V_N)]$.

(c) (4 points) Consider an N step model (S_n) (bank account $S^0 = 1$) of conditional binomial form, i.e. the parameters of the model dependent on a fixed two state Markov chain (X_n),

$$P[S_{n+1} = S_n u | S_n, X_n] = p(X_n); P[S_{n+1} = S_n d | S_n, X_n] = 1 - p(X_n) \ ,$$

where $u > 1 > d > 0$ and an exponential utility function $U(x) = 1 - \exp(-x)$. Given a utility u write down in pseudocode an algorithm that learns the self-financing trading strategy with initial capital x for $T = N$ trading days using M generated trajectories of the binomial market model. For the training part, use stochastic gradient descent with step size $\gamma > 0$ and make $K \in \mathbb{N}$ epochs. Gradients are computed with mini-batches of size one.

Hint: You may use a function `phi(x,`θ`)` that implements a smooth neural network with input `x`$\in \mathbb{R}^n$ where you can specify the input dimension n before using the function and $\theta \in \mathbb{R}^L$ is a parameter vector corresponding to the weights of the neural network. You do not need to specify an architecture or say what L is, you can assume `phi` is suitable for approximating any continuous function. If you need multiple neural networks, write them as `phi`$_i$ for $i \in \mathbb{N}$. In that case, specify each input dimension with n_i, the corresponding weights with θ_i.

The trajectories are stored in a Matrix `S` $\in \mathbb{R}^{M \times (N+1)}$.

Finally, you can use derivatives such as gradients at will. E.g. if you have a value `cost`, that depends in some way on the values of θ, then `grad =`∇_θ `cost` computes the gradient w.r.t. the weights θ and stores the gradient in `grad` for the current values of θ. If the values of θ change, you may assume that the value of `grad` changes automatically to the gradient evaluated at the new value of θ.

Question 3. (16 points)

(a) Calibration of models: describe the calibration problem as inverse problem of selecting a model given some data and some pool of models.

[2 Points]

(b) Describe the Bayesian approach of model selection and compare it to the optimization approach. Explain in particular the connection of the choice of a prior and regularization. Why is 'gradient descent plus noise' a good approach for training?

[4 Points]

(c) What is a local volatility model and what are they for: derive the relationship between option prices and the local volatility function.

[2 Points]

(d) What is a local stochastic volatility model and what are they for: why does this not lead to a standard stochastic differential equation?

[2 Points]

(e) Write down pseudocode to learn a local stochastic volatility for finitely many given option prices: assume a Heston stochastic variance and parametrize local volatility by a neural network and solve the pricing equation by an Euler scheme, then define a loss function and write down the optimization problem that one needs to solve such that model prices and market prices are close. Discuss mini-batching in this approach, does it work or rather not?

[6 Points]

Question 4. (10 points)

(a) (3 points) Machine Learning in Finance:

i. When could it be useful to try a machine learning approach in Finance?

A. For example for the Black Scholes Formula: we can train a neural network to approximate its values.

B. When we do not want to use a stochastic model for the market.

C. When the problem is mathematically well specified, but difficult to solve with classical numerical techniques, e.g. high dimensional problems with transaction costs.

D. When we do not have an idea how to formulate the problem mathematically.

ii. Which typical problem for finance has never been treated by machine learning techniques?

A. The hedging problem, i.e. how to invest in a financial market to reduce risks.

B. The calibration problem, i.e. how to choose a model from a pool of models.

C. The simulation problem, i.e. how to generate realistic scenarios for a financial market artficially.

D. The risk measurement problem, i.e. how to choose a risk measure correctly.

iii. Which main ingredients do you need to formulate a machine learning problem in finance?

A. A neural network encoding actions in a financial market, a loss function and a training method.

B. A stochastic model, a utility function and a set of portfolio strategies.

C. A stochastic model, a loss function and a portfolio strategy.

D. A neural network, a utility function and an idea about risk aversion.

(b) (3 points) Architectures:

i. Which of the following terms does not describe an architecture of a neural network?

A. Recurrent networks.

B. Resilient networks.

C. Residual networks.

D. Convolutional networks.

ii. Do neural networks, which work well in practice, depend on many parameters?

A. No, it is an important principle in science to use as few parameters as possible.

B. No, neural networks only use affine functions which depend on few parameters.

C. Yes, since Relu depends on many parameters.

D. Yes, since the affine functions in each layer map high dimensional spaces to high dimensional spaces and do therefore need high dimensional matrices.

iii. Consider a shallow network with k hidden nodes on $[0,1]^d$ mapping to $\mathbb{R}$ (we always take a shift term here).

A. The number of parameters is $kd + k$ for the affine function in the hidden layer and $k + 1$ for the last layer.

B. The number of parameters is kd for the affine function in the hidden layer and $k + 1$ for the last layer.
C. The number of parameters is $kd + k$ for the affine function in the hidden layer and k for the last layer.
D. The number of parameters is kd for the affine function in the hidden layer and k for the last layer.

(c) (4 points) Describe in a short text which problems can be treated by Machine Learning in Finance and why this might be important for future developments in the financial sector. Which problems do you expect to be treated in the future?