

Національний технічний університет України «КПІ ім. Ігоря Сікорського»  
Факультет Інформатики та Обчислювальної Техніки  
Кафедра інформаційних систем та технологій

**Лабораторна робота № 6**  
з дисципліни «Методи та технології штучного інтелекту»  
На тему:  
«Нейро-нечітке моделювання»

Виконала:  
студентка групи ІС-12.  
Мельникова К.О.

Перевірив:  
Шимкович В. М.

**Мета роботи:** отримання і закріплення знань про методи моделювання та принципи функціонування нейронечітких систем, а також формування практичних навичок з конструювання нейронечітких мереж).

**Завдання.**

1. Сформулювати завдання в галузі обчислювальної техніки, для вирішення якої було б обґрунтовано застосування гібридної нейронечіткої мережі.
2. Сформувати вибірку для навчання гібридної нейронної мережі.
3. Згенерувати і візуалізувати структуру гібридної нейронної мережі.
4. Навчити гібридну нейронну мережу, при цьому задати і обґрунтувати параметри її навчання.
5. Побудувати систему нечіткого виводу для отриманої гібридної нейронної мережі.
6. Виконати перевірку адекватності побудованої нечіткої моделі гібридної мережі.
7. Оформіть звіт по лабораторній роботі.

- 1) Для виконання даної лабораторної роботи згенеруємо нейронну мережу, що передбачатиме курс біткоїна на основі даних за минулий рік.

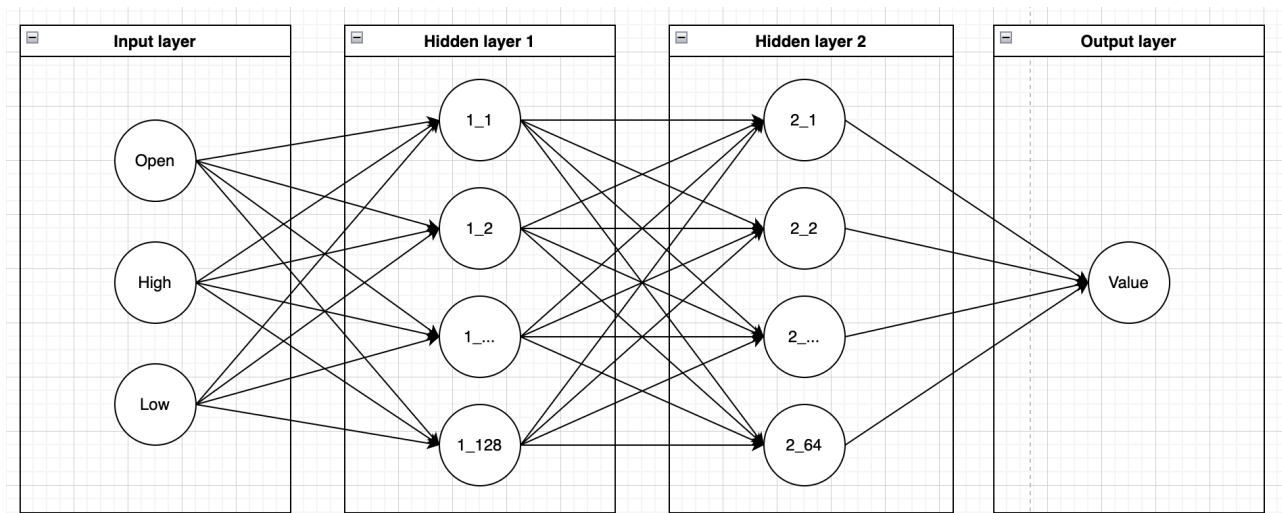
```
Date,Open,High,Low,Close,Adj Close,Volume
2022-12-14,17782.066406,18318.531250,17739.513672,17815.650391,17815.650391,25534481470
2022-12-15,17813.644531,17846.744141,17322.589844,17364.865234,17364.865234,20964448341
2022-12-16,17364.546875,17505.525391,16584.701172,16647.484375,16647.484375,24031608960
2022-12-17,16646.982422,16800.589844,16614.029297,16795.091797,16795.091797,14463581825
2022-12-18,16795.609375,16815.386719,16697.820313,16757.976563,16757.976563,10924354698
2022-12-19,16759.041016,16807.527344,16398.136719,16439.679688,16439.679688,17221074814
2022-12-20,16441.787109,17012.984375,16427.867188,16906.304688,16906.304688,22722096615
2022-12-21,16904.527344,16916.800781,16755.912109,16817.535156,16817.535156,14882945045
2022-12-22,16818.380859,16866.673828,16592.408203,16830.341797,16830.341797,16441573050
2022-12-23,16829.644531,16905.218750,16794.458984,16796.953125,16796.953125,15329265213
2022-12-24,16796.976563,16864.703125,16793.527344,16847.755859,16847.755859,9744636213
2022-12-25,16847.505859,16860.554688,16755.253906,16841.986328,16841.986328,11656379938
2022-12-26,16842.250000,16920.123047,16812.369141,16919.804688,16919.804688,11886957804
2022-12-27,16919.291016,16959.845703,16642.072266,16717.173828,16717.173828,15748580239
2022-12-28,16716.400391,16768.169922,16497.556641,16552.572266,16552.572266,17005713920
2022-12-29,16552.322266,16651.755859,16508.683594,16642.341797,16642.341797,14472237479
2022-12-30,16641.330078,16643.427734,16408.474609,16602.585938,16602.585938,15929162910
2022-12-31,16603.673828,16628.986328,16517.519531,16547.496094,16547.496094,11239186456
```

Для виконання поставленої задачі буде достатньо лише параметрів Date, Open, High, Low та Close.

Тому, для подальшої роботи спершу підготуємо датасет:

```
data = pd.read_csv('BTC-USD.csv')
data["Date"] = pd.to_datetime(data["Date"])
data = data.sort_values(by=["Date"]) # відсортуємо по даті
print(data)
X = data[['Open', 'High', 'Low']] # значення
y = data['Close'] # мітки
# перетворимо дані на тензори зі значеннями типу float
X, y = torch.tensor(X.values).float(), torch.tensor(y.values).float()
# розділимо дані на тестовий та навчальний датасети
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=len(data)-1, shuffle=False)
y_train, y_test = y_train.unsqueeze(1), y_test.unsqueeze(1)
```

- 2) Визначимо структуру та параметри нейронної мережі. Так як ми маємо три параметри, відповідно на вхід буде задано три нейрони. Для виконання даної лабораторної роботи буде достатньо 3-х шарів: перший на вхід матиме 3, на вихід 128 нейрони, другий – 128 та 64, та третій, вихідний шар братиме 64 та на вихід даватиме один нейрон.



Визначимо параметри мережі та створимо клас нашої нейронної мережі:

```
epochs_number = 500
first_layer_nn = 128
second_layer_nn = 64

class MyNN(nn.Module): # пронаслідуюмо клас нашої нейронної мережі від nn.Module
    def __init__(self):
        super().__init__()
        self.layer1 = nn.Linear(3, first_layer_nn) # визначимо шари
        self.layer2 = nn.Linear(first_layer_nn, second_layer_nn)
        self.layer3 = nn.Linear(second_layer_nn, 1)
    def forward(self, x): # функція руху по мережі
        x = f.relu(self.layer1(x))
        x = f.relu(self.layer2(x))
        x = self.layer3(x)
        return x

model = MyNN()
```

Виконаємо навчання мережі за допомогою методу стохастичного градієнтного спуску з оптимізатором Adam та використанням середньоквадратичної функції втрат для вимірювання помилок:

```
lossF = nn.MSELoss() # функція втрат
optimize = optim.Adam(model.parameters(), lr=0.001) # Використаємо оптимізатор Adam
trainLosses, testLosses = [], []
for epoch in range(epochs_number): # тренування мережі
    model.train()
    optimize.zero_grad() # обнуляємо градієнти
    pred = model(X_train) # прямий прохід (forward pass) моделі для тренувальних даних
    loss = lossF(pred, y_train) # обчислимо втрати за допомогою середньоквадратичної функції втрат
    loss.backward() # Обчислимо градієнти за допомогою backpropagation
    optimize.step() # оновлення ваг на основі градієнтного спуску
    trainLosses.append(loss.item())
    model.eval()
    with torch.no_grad(): # Оцінка моделі на тестових даних
        pred = model(X_test)
        testLoss = lossF(pred, y_test)
        testLosses.append(testLoss.item())
    print(f'Epoch {epoch}, Loss: {loss.item():.2f}, Test Loss:{testLoss.item():.2f}')
```

```
Epoch 0, Loss: 288633472.00, Test Loss:616925440.00
Epoch 1, Loss: 245333664.00, Test Loss:526700096.00
Epoch 2, Loss: 208858160.00, Test Loss:450216864.00
Epoch 3, Loss: 178175328.00, Test Loss:379161280.00
Epoch 4, Loss: 149666256.00, Test Loss:316001248.00
Epoch 5, Loss: 124405816.00, Test Loss:260374464.00
Epoch 6, Loss: 102152120.00, Test Loss:209981584.00
Epoch 7, Loss: 82047416.00, Test Loss:165626848.00
Epoch 8, Loss: 64281636.00, Test Loss:127808272.00
Epoch 9, Loss: 49271516.00, Test Loss:94756520.00
Epoch 10, Loss: 36195324.00, Test Loss:66621772.00
Epoch 11, Loss: 25125832.00, Test Loss:43534176.00
Epoch 12, Loss: 16107383.00, Test Loss:25966440.00
Epoch 13, Loss: 9331480.00, Test Loss:13395822.00
Epoch 14, Loss: 4561486.50, Test Loss:5170243.50
Epoch 15, Loss: 1549315.00, Test Loss:956910.81
Epoch 16, Loss: 155236.00, Test Loss:210458.70
Epoch 17, Loss: 146037.42, Test Loss:2041093.62
Epoch 18, Loss: 1155977.75, Test Loss:5630989.50
Epoch 19, Loss: 2851672.00, Test Loss:10134755.00
Epoch 20, Loss: 4884126.00, Test Loss:14726545.00
Epoch 21, Loss: 6913417.50, Test Loss:18700360.00
Epoch 22, Loss: 8650032.00, Test Loss:21549348.00
Epoch 23, Loss: 9887328.00, Test Loss:23003576.00
Epoch 24, Loss: 10516834.00, Test Loss:23022604.00
...
Epoch 496, Loss: 0.00, Test Loss:210367.55
Epoch 497, Loss: 0.00, Test Loss:210367.55
Epoch 498, Loss: 0.00, Test Loss:210367.55
Epoch 499, Loss: 0.00, Test Loss:210367.55
```

Побудуємо графіки:

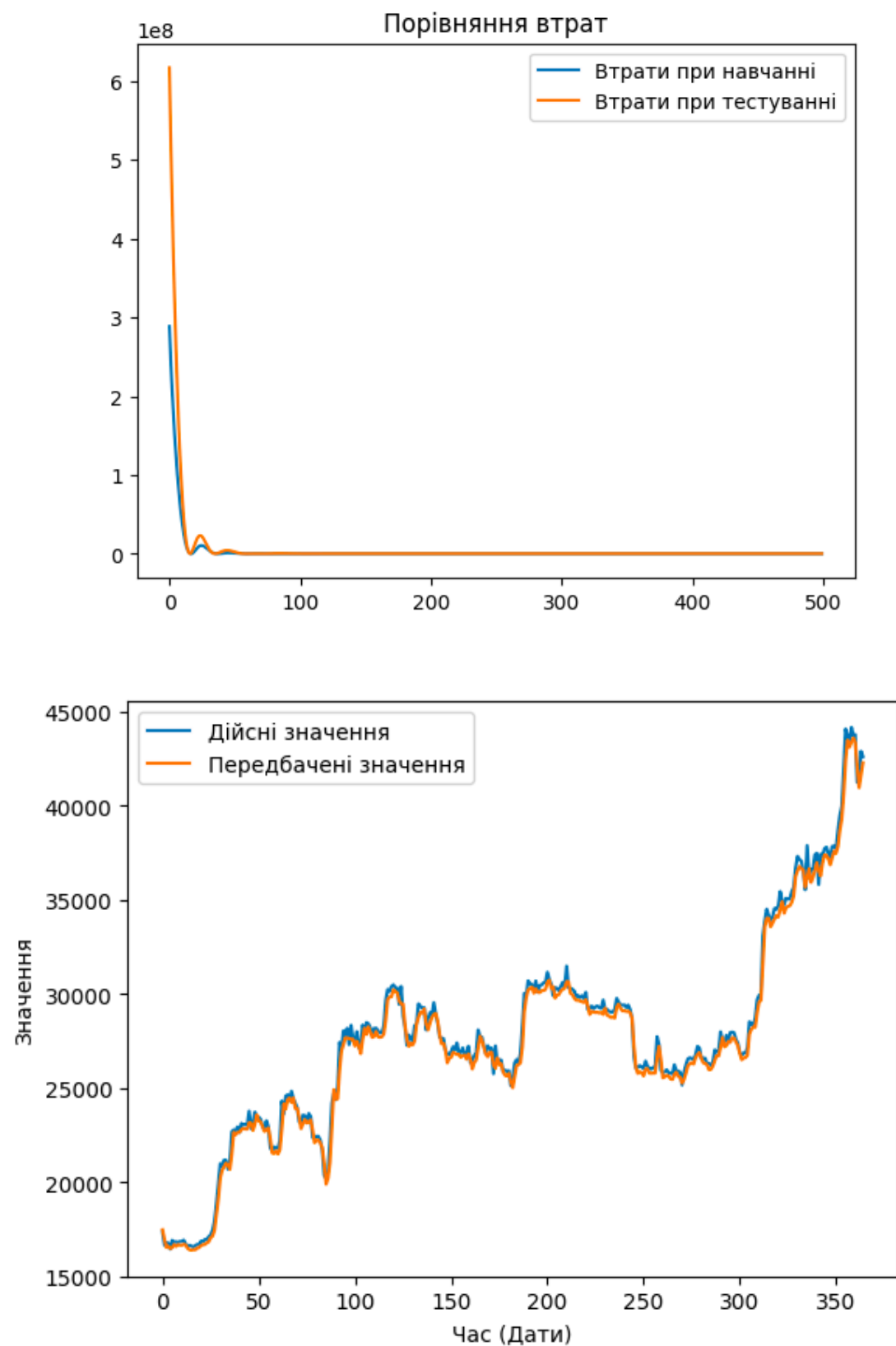
```
plt.plot(trainLosses, label='Втрати при навчанні')
plt.plot(testLosses, label='Втрати при тестуванні')
plt.title('Порівняння втрат')
plt.legend()
plt.show()

model.eval()
with torch.no_grad():
    pred = model(X_test)
plt.plot(y_test.numpy(), label='Дійсні значення')
plt.plot(pred.numpy(), label='Передбачені значення')
plt.xlabel('Час (Дати)')
plt.ylabel('Значення')
plt.legend()
plt.show()

forecastX = data.iloc[-1][['Open', 'High', 'Low']].astype(float).values
forecastX = torch.tensor(forecastX).float()
model.eval()
with torch.no_grad():
    forecastY = model(forecastX.unsqueeze(0)) # спрогнозуємо значення

predicted_price = forecastY.item()
real_price = data.iloc[-1]["Close"]
print(f'Передбачена ціна за {data.iloc[-1]["Date"]}: {predicted_price}')
print(f'Дійсна ціна за {data.iloc[-1]["Date"]}: {real_price}')
print(f'Різниця: {abs(predicted_price-real_price):.6f}')
```

✓ 0.1s



Передбачена ціна за 2023-12-14 00:00:00:42284.14453125  
Дійсна ціна за 2023-12-14 00:00:00: 42607.230469  
Різниця: 323.085938

## **Висновок.**

Отже, у даній роботі я отримала і закріпила знання про методи моделювання та принципи функціонування нейронечітких систем, а також сформулювала практичні навички з конструювання нейронечітких мереж.

У ході виконання лабораторної роботи я створила мережу, що коректно прогнозує тренди та доволі близько – значення курсу криптовалюти Bitcoin.

Результати отримані у ході виконання роботи проілюстровані у текстовому форматі та у вигляді графіків підтверджують те, що модель працює досить непогано. Хоч значення і відрізняються, насправді вони є дуже близькими до істинних. Тож модель працює успішно.