

Додаток А

Графічні матеріали

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

ЗАТВЕРДЖУЮ

Керівник атестаційної роботи

_____ доц. В.М. Решетнік
(підпис)

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ
УПРАВЛІННЯ ПАЛИВНО-ЕНЕРГЕТИЧНИМ БАЛАНСОМ РЕГІОНУ

Графічний матеріал

ЛИСТ ЗАТВЕРДЖЕННЯ
ГЮІК.506190.007 ПЗ - 01 12 01- ЛУ

Листів 18

Студент групи ІТПм-18-1

_____ Пуговкін М.А.
(підпис)

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

ЗАТВЕРДЖЕНО

ГЮІК. 506190.007 ПЗ - 01 12 01-ЛУ

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ
УПРАВЛІННЯ ПАЛИВНО-ЕНЕРГЕТИЧНИМ БАЛАНСОМ РЕГІОНУ

Графічний матеріал

ГЮІК. 506190.007 ПЗ - 01 12 01- ЛУ

Листів 18

2019

АКТУАЛЬНІСТЬ

Недоліки існуючих систем:

- відсутня функція подання зведеного ПЕБ поточного року;
- відсутня функція контролю за надходженнями та витратами всіх ПЕР (наявний контроль лише за природнім газом та електроенергією);
- відсутня функція формування прогнозних оцінок ПЕБ на наступні 3 роки.



Розроб.	Пуговкін М.А.			<i>Розробка та дослідження інформаційної системи управління ПЕБ регіону</i>	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				<i>ІТІМ-18-1</i>	<i>Лист 1</i>
Затверд.	Гребеннік І.В.			<i>СТ</i>	<i>Листів 18</i>

АКТУАЛЬНІСТЬ

2

Актуальність даної атестаційної роботи зумовлена тим, що проблеми науково-обґрунтованого автоматизованого формування паливно-енергетичного балансу регіону в умовах забезпечення енергетичної безпеки держави та диверсифікації енергетичного є вкрай важливим питанням сьогодення.



Розроб.	Пуговкін М.А.			<i>Розробка та дослідження інформаційної системи управління ПЕБ регіону</i>	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				<i>ІТІМ-18-1</i>	<i>Лист 2</i>
Затверд.	Гребеннік І.В.			<i>СТ</i>	<i>Листів 18</i>

3

МЕТА

РОБОТИ

розробка ІСУ ПЕБ регіону з базовою функцією автоматизованого формування ПЕБ регіону та додатковою функцією прогнозування ПЕБ регіону

ОБ'ЄКТ

РОЗРОБКИ

процеси інформаційного забезпечення системи управління ПЕБ регіону

ПРЕДМЕТ

РОЗРОБКИ

інформаційна система управління ПЕБ регіону



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІІІМ-18-1	Лист 3
Затверд.	Гребеннік І.В.			СТ	Листів 18

ПОСТАНОВКА ЗАДАЧІ

Виходячи з проведеного аналізу предметної області та виявлених недоліків існуючих рішень, метою роботи є розробка і дослідження інформаційної системи управління для автоматизації формування паливно-енергетичного балансу регіону. Для досягнення цієї мети необхідно розв'язати такі завдання:



Розроб.	Пуговкін М.А.			<i>Розробка та дослідження інформаційної системи управління ПЕБ регіону</i>	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				<i>ІТІм-18-1</i>	<i>Лист 4</i>
Затверд.	Гребеннік І.В.			<i>СТ</i>	<i>Листів 18</i>

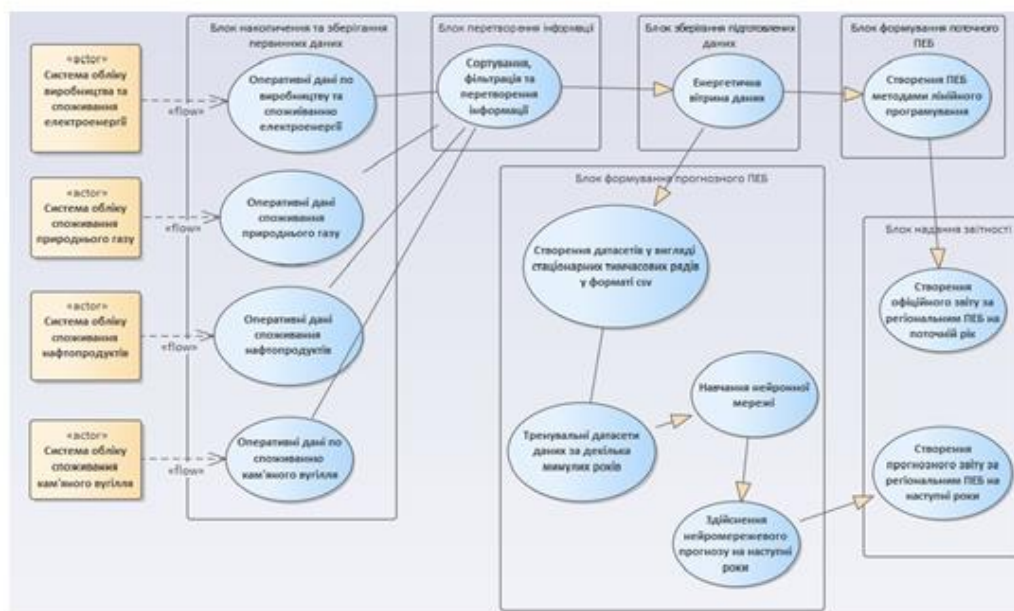
ПОСТАНОВКА ЗАДАЧІ

- розробити системні та функціональні вимоги до компонентів ІСУ;
- виконати математичну формалізацію та створити алгоритм розв'язання задачі розрахунку паливно-енергетичного балансу на основі задачі лінійного програмування;
- спроектувати модуль прогнозування паливно-енергетично балансу на основі штучної нейронної мережі;
- обґрунтувати вибір параметрів для штучної нейронної мережі до модулю прогнозування;
- розробити відповідне програмне забезпечення ІСУ;
- провести дослідження модуля прогнозування ІСУ.



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІТІМ-18-1	Лист 5
Затверд.	Гребеннік І.В.			СТ	Листів 18

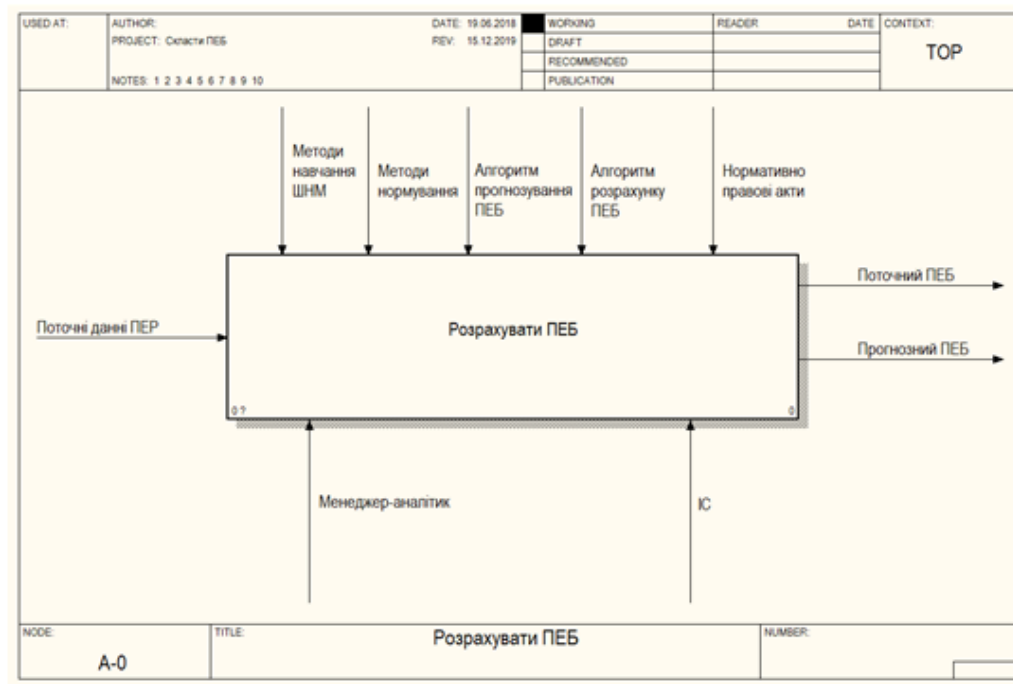
ЗАГАЛЬНА ДІАГРАМА ПРЕЦЕДЕНТІВ ІСУ ПЕБ РЕГІОНУ 6



Розроб.	Луговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІТІМ-18-1	Лист 6
Затверд.	Гребеннік І.В.			СТ	Листів 18

КОНТЕКСТНА ДІАГРАМА СИСТЕМИ

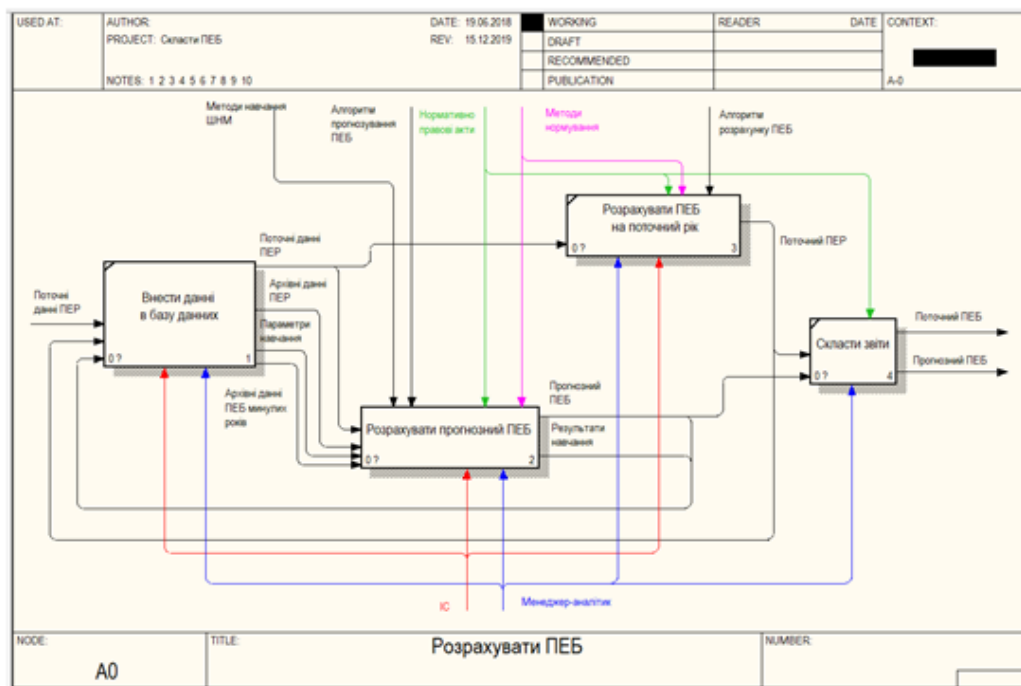
7



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІІІІм-18-1	Лист 7
Затверд.	Гребеннік І.В.			СТ	Листів 18

ДЕКОМПОЗИЦІЯ КОНТЕКСТНОЇ ДІАГРАМИ

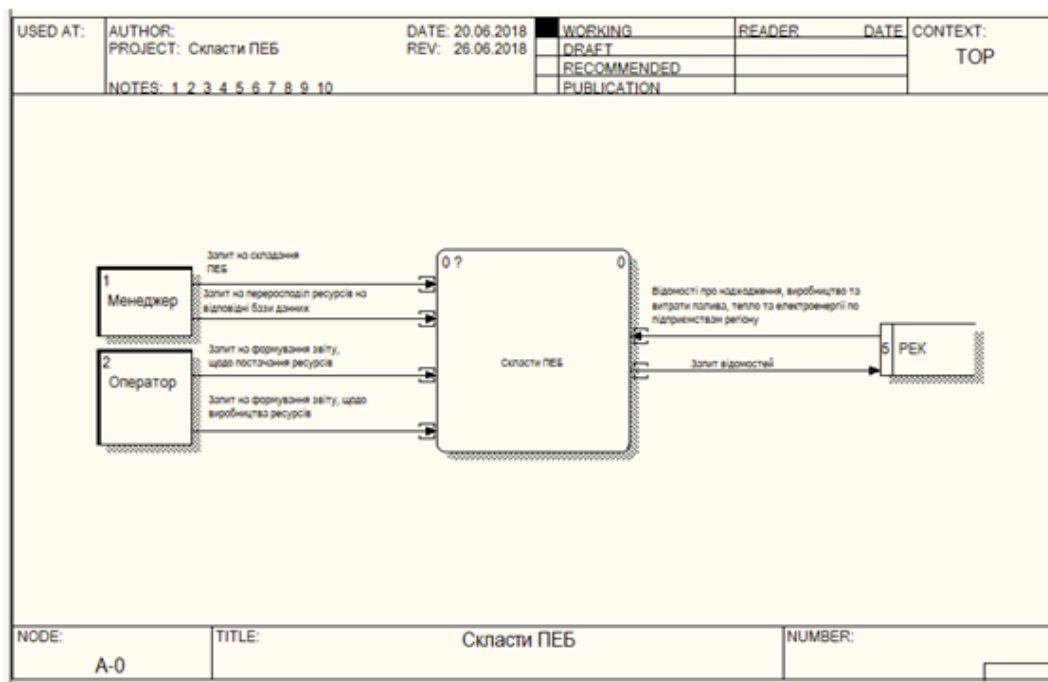
8



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІТІПм-18-1	Лист 8
Затверд.	Гребеннік І.В.			СТ	Листів 18

ДИАГРАММА ПОТОКОВ ДАНИХ

9

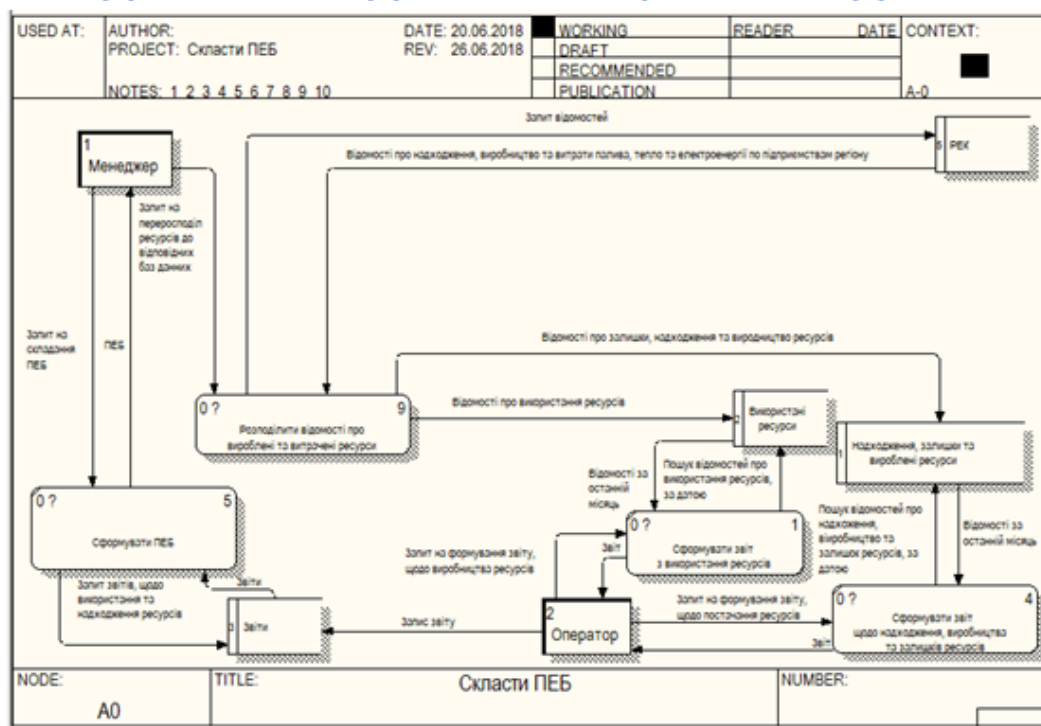


NODE: A-0	TITLE: Скласти ПЕБ	NUMBER:
--------------	-----------------------	---------



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону
Перевір.	Решетнік В.М.			
Н. Контр.	Решетнік В.М.			
				ІТІМ-18-1
				СТ
Затверд.	Гребеннік І.В.			Лист 9
				Листів 18

ДИАГРАММА ДЕКОМПОЗИЦІЇ ПОТОКІВ ДАНИХ 10



Розроб.	Пуговкін М.А.			<i>Розробка та дослідження інформаційної системи управління ПЕБ регіону</i>
Перевір.	Решетнік В.М.			
Н. Контр.	Решетнік В.М.			
			ІТІМ-18-1	Лист 10
Затверд.	Гребеннік І.В.		СТ	Листів 18

МАТЕМАТИЧНІ МЕТОДИ РОЗРАХУНКУ ПОТОЧНОГО ПЕБ

12

Загальним завданням ЛП називається завдання, яке полягає в знаходженні максимального (мінімального) значення функції

Загальний вигляд:

$$F = \sum_{j=1}^n c_j x_j \quad (2.1)$$

За умови

Де a_{ij} , b_i , c_j – задані постійні величини k та n .

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, i = \overline{1, 2, \dots, k} \quad (2.2)$$

Функція (2.1) називається цільовою функцією задачі (2.2), а умова (2.2) – обмеженням цієї задачі.



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІТІМ-18-1	Лист 12
Затверд.	Гребеннік І.В.			СТ	Листів 18

МАТЕМАТИЧНІ МЕТОДИ РОЗРАХУНКУ ПОТОЧНОГО ПЕБ 13

Особливістю стандартної задачі ЛП є те, що її обмеження представлені у вигляді лінійних нерівностей, а також умов невід'ємності змінних, присутніх в задачі:

$$\sum_{j=1}^n c_j x_j \rightarrow \max \quad (2.3)$$

$$\begin{cases} \sum_{j=1}^n a_{ij} x_j \leq b_i & i = 1, 2, \dots, n \\ x_j \geq 0 & j = 1, 2, \dots, n \end{cases} \quad (2.4)$$

На основі записаних вище формул, розв'язано задачу розрахунку поточного ПЕБ методами лінійного програмування.



Розроб.	Пуговкін М.А.			<i>Розробка та дослідження інформаційної системи управління ПЕБ регіону</i>	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				<i>ІІІМ-18-1</i>	<i>Лист 13</i>
Затверд.	Гребеннік І.В.			<i>СТ</i>	<i>Листів 18</i>

РЕЗУЛЬТАТИ ОПТИМІЗАЦІЇ

14

Вид палива	Q_p^1 , ккал/кг ккал/нм ³	Питомі затрати грн/т грн/тис.нм ³	Використано 01.19. на	Загрузка 01.19. на	Сум. витрати на 01.19.
Шлам	2400	48,65	39552 т		
Бур. вугілля	2100	130,5	200 т		
Прир. газ	7990	413,61	17930 тис.нм ³	279787,4 Гкал.	13731436 грн.
Супут. газ	9130	593,95	3270 тис.нм ³		
Мазут	9850	2106,86	1150 т		

Вхідні дані

Вид палива	Оптимальні витрати.	Сум. загрузка Гкал	Сум. витрати грн
Шлам	35000 т	279787,6	11836200
Бур. Вугілля	0		
Прир. Газ	24500 тис.нм ³		
Супут. Газ	0		
Мазут	0		

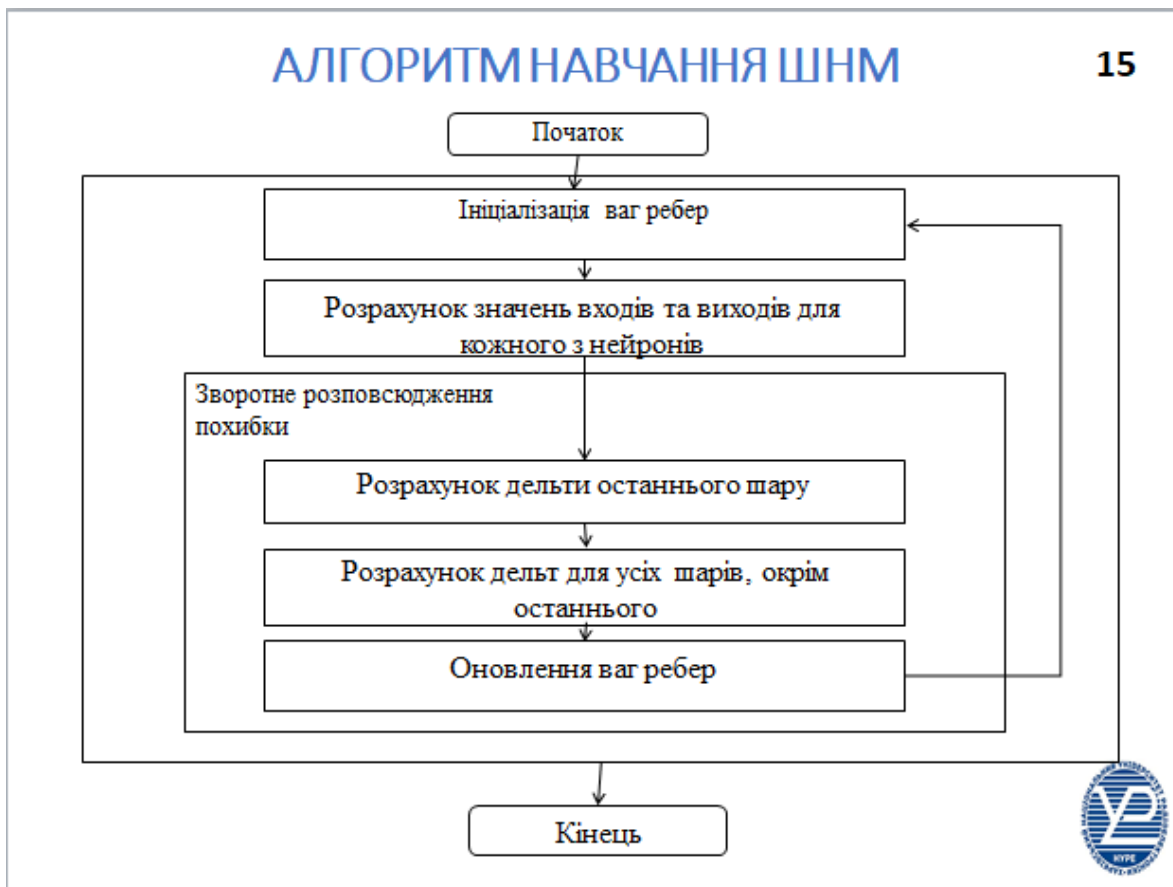
Результат оптимізації



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІТІМ-18-1	Лист 14
Затверд.	Гребеннік І.В.			СТ	Листів 18

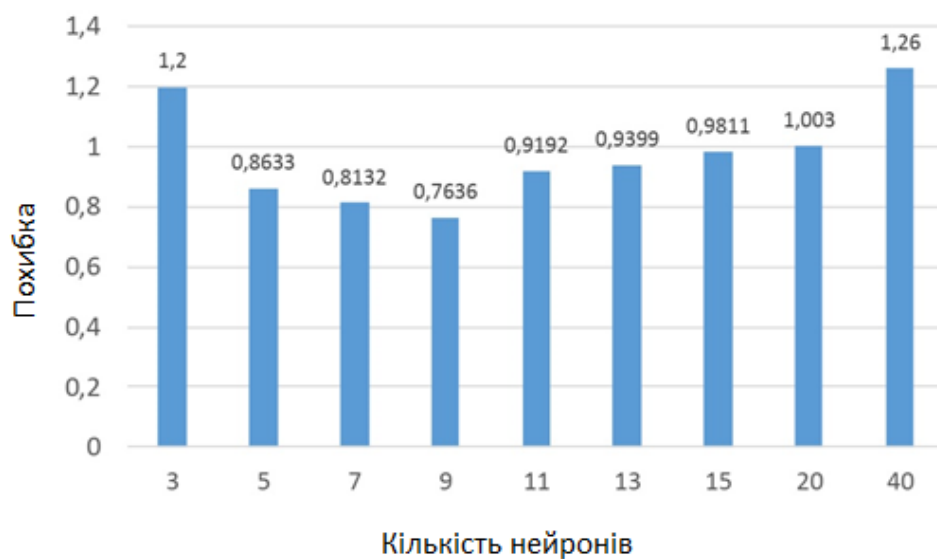
АЛГОРИТМ НАВЧАННЯ ШНМ

15



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІТІМ-18-1	Лист 15
Затверд.	Гребеннік І.В.			СТ	Листів 18

ЕФЕКТИВНІСТЬ АЛГОРИТМУ НАВЧАННЯ ШНМ 16



Залежність похибки від кількості нейронів в прихованому шарі



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІТІМ-18-1	Лист 16
Затверд.	Гребеннік І.В.			СТ	Листів 18

ВИСНОВКИ

17

В результаті виконання магістерської роботи розроблена інформаційної системи управління паливно-енергетичним балансом регіону

На етапі підготовки до проектування проаналізовано предметну область, порівняні вже існуючі реалізовані системи, виявлені їх переваги та недоліки, також виявлені проблеми та актуалізовані рішення щодо їх подолання. Спроектвані різні аспекти інформаційної системи, а також її архітектура.

На етапі розробки інформаційної системи управління паливно-енергетичним балансом регіону розроблено серверну та клієнтську частини додатку. Розроблені модуль формування поточного та прогнозного ПЕБ.



Розроб.	Луговкін М.А.			<i>Розробка та дослідження інформаційної системи управління ПЕБ регіону</i>	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				<i>ІТІМ-18-1</i>	<i>Лист 17</i>
Затверд.	Гребеннік І.В.			<i>СТ</i>	<i>Листів 18</i>

ВИСНОВКИ

18

Проведені дослідження ефективності модуля прогнозування шляхом емпіричного підбору значення таких гіперпараметрів штучної нейронної мережі, як швидкість навчання, момент навчання і кількість нейронів прихованого шару.

Були виявлені наступні оптимальні значення гіперпараметрів нейронної мережі:

- а) коефіцієнт навчання = 0.03;
- б) кількість нейронів в прихованому шарі $n = 7$;
- в) момент навчання $m = 0.5$.

У разі використання цих гіперпараметрів значення середньоквадратичного відхилення буде найменшим.



Розроб.	Пуговкін М.А.			Розробка та дослідження інформаційної системи управління ПЕБ регіону	
Перевір.	Решетнік В.М.				
Н. Контр.	Решетнік В.М.				
				ІТІМ-18-1	Лист 18
Затверд.	Гребеннік І.В.			СТ	Листів 18

Додаток Б

Текст програми

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

ЗАТВЕРДЖУЮ
Керівник атестаційної роботи
_____доц. В.М. Решетнік
(підпис)

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ
ПАЛИВНО-ЕНЕРГЕТИЧНИМ БАЛАНСОМ РЕГІОНУ

Текст програми

ЛИСТ ЗАТВЕРДЖЕННЯ
ГЮІК.506190.007 ПЗ - 01 12 01- ЛУ

Студент групи ІТІм-18-1
_____ Пуговкін М.А.
(підпис)

2019

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

ЗАТВЕРДЖЕНО

ГЮІК. 506190.007 ПЗ - 01 12 01-ЛУ

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ
ПАЛИВНО-ЕНЕРГЕТИЧНИМ БАЛАНСОМ РЕГІОНУ

Текст програми

ГЮІК. 506190.007 ПЗ - 01 12 01- ЛУ

Аркушів 6

2019

```

import torch
import torch.nn as nn
from torch.utils.data import DataLoader
import torchvision.transforms as transforms
import torchvision.datasets
from bokeh.plotting import figure
from bokeh.io import show
from bokeh.models import LinearAxis, Range1d
import numpy as np

# Hyperparameters
num_epochs = 6
num_classes = 10
batch_size = 100
learning_rate = 0.001

DATA_PATH = '\PycharmProjects\MNISTData'
MODEL_STORE_PATH = '\PycharmProjects\pytorch_models\'

# transforms to apply to the data
trans = transforms.Compose([transforms.ToTensor(), transforms.Normalize((0.1307,),
(0.3081,))])

# MNIST dataset
train_dataset = torchvision.datasets.MNIST(root=DATA_PATH, train=True,
transform=trans, download=True)
test_dataset = torchvision.datasets.MNIST(root=DATA_PATH, train=False,
transform=trans)

# Data loader
train_loader = DataLoader(dataset=train_dataset, batch_size=batch_size, shuffle=True)
test_loader = DataLoader(dataset=test_dataset, batch_size=batch_size, shuffle=False)

# Convolutional neural network (two convolutional layers)
class ConvNet(nn.Module):
    def __init__(self):
        super(ConvNet, self).__init__()
        self.layer1 = nn.Sequential(
            nn.Conv2d(1, 32, kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.layer2 = nn.Sequential(
            nn.Conv2d(32, 64, kernel_size=5, stride=1, padding=2),
            nn.ReLU(),
            nn.MaxPool2d(kernel_size=2, stride=2))
        self.drop_out = nn.Dropout()

```

```

self.fc1 = nn.Linear(7 * 7 * 64, 1000)
self.fc2 = nn.Linear(1000, 10)

def forward(self, x):
    out = self.layer1(x)
    out = self.layer2(out)
    out = out.reshape(out.size(0), -1)
    out = self.drop_out(out)
    out = self.fc1(out)
    out = self.fc2(out)
    return out

model = ConvNet()

# Loss and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

# Train the model
total_step = len(train_loader)
loss_list = []
acc_list = []
for epoch in range(num_epochs):
    for i, (images, labels) in enumerate(train_loader):
        # Run the forward pass
        outputs = model(images)
        loss = criterion(outputs, labels)
        loss_list.append(loss.item())

        # Backprop and perform Adam optimisation
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

        # Track the accuracy
        total = labels.size(0)
        _, predicted = torch.max(outputs.data, 1)
        correct = (predicted == labels).sum().item()
        acc_list.append(correct / total)

    if (i + 1) % 100 == 0:
        print('Epoch [{}/{}], Step [{}/{}], Loss: {:.4f}, Accuracy: {:.2f}%'.format(
            epoch + 1, num_epochs, i + 1, total_step, loss.item(),
            (correct / total) * 100))

# Test the model
model.eval()
with torch.no_grad():

```

```

correct = 0
total = 0
for images, labels in test_loader:
    outputs = model(images)
    _, predicted = torch.max(outputs.data, 1)
    total += labels.size(0)
    correct += (predicted == labels).sum().item()

print("Test Accuracy of the model on the 10000 test images: { } %".format((correct /
total) * 100))

# Save the model and plot
torch.save(model.state_dict(), MODEL_STORE_PATH + 'conv_net_model.ckpt')

p = figure(y_axis_label='Loss', width=850, y_range=(0, 1), title='PyTorch ConvNet
results')
p.extra_y_ranges = {'Accuracy': Range1d(start=0, end=100)}
p.add_layout(LinearAxis(y_range_name='Accuracy', axis_label='Accuracy (%)'), 'right')
p.line(np.arange(len(loss_list)), loss_list)
p.line(np.arange(len(loss_list)), np.array(acc_list) * 100, y_range_name='Accuracy',
color='red')
show(p)

def convert_y_to_vect(y):
    y_vect = np.zeros((len(y), 10))
    for i in range(len(y)):
        y_vect[i, y[i]] = 1
    return y_vect

def f(x):
    return 1 / (1 + np.exp(-x))

def f_deriv(x):
    return f(x) * (1 - f(x))

def setup_and_init_weights(nn_structure):
    W = {}
    b = {}
    for l in range(1, len(nn_structure)):
        W[l] = r.random_sample((nn_structure[l], nn_structure[l-1]))
        b[l] = r.random_sample((nn_structure[l],))
    return W, b

def init_tri_values(nn_structure):
    tri_W = {}

```

```

tri_b = {}
for l in range(1, len(nn_structure)):
    tri_W[l] = np.zeros((nn_structure[l], nn_structure[l-1]))
    tri_b[l] = np.zeros((nn_structure[l],))
return tri_W, tri_b

```

```

def feed_forward(x, W, b):
    h = {1: x}
    z = {}
    for l in range(1, len(W) + 1):
        # if it is the first layer, then the input into the weights is x, otherwise,
        # it is the output from the last layer
        if l == 1:
            node_in = x
        else:
            node_in = h[l]
        z[l+1] = W[l].dot(node_in) + b[l] #  $z^{(l+1)} = W^{(l)}h^{(l)} + b^{(l)}$ 
        h[l+1] = f(z[l+1]) #  $h^{(l)} = f(z^{(l)})$ 
    return h, z

```

```

def calculate_out_layer_delta(y, h_out, z_out):
    #  $\delta^{(nl)} = -(y_i - h_i^{(nl)}) * f'(z_i^{(nl)})$ 
    return -(y-h_out) * f_deriv(z_out)

```

```

def calculate_hidden_delta(delta_plus_1, w_1, z_1):
    #  $\delta^{(l)} = (\text{transpose}(W^{(l+1)}) * \delta^{(l+1)}) * f'(z^{(l)})$ 
    return np.dot(np.transpose(w_1), delta_plus_1) * f_deriv(z_1)

```

```

def train_nn(nn_structure, X, y, iter_num=3000, alpha=0.25):
    W, b = setup_and_init_weights(nn_structure)
    cnt = 0
    m = len(y)
    avg_cost_func = []
    print('Starting gradient descent for {} iterations'.format(iter_num))
    while cnt < iter_num:
        if cnt%1000 == 0:
            print('Iteration {} of {}'.format(cnt, iter_num))
            tri_W, tri_b = init_tri_values(nn_structure)
            avg_cost = 0
            for i in range(len(y)):
                delta = {}
                # perform the feed forward pass and return the stored h and z values, to be used in
                the
                # gradient descent step
                h, z = feed_forward(X[i, :], W, b)

```

```

# loop from nl-1 to 1 backpropagating the errors
for l in range(len(nn_structure), 0, -1):
    if l == len(nn_structure):
        delta[l] = calculate_out_layer_delta(y[i,:], h[l], z[l])
        avg_cost += np.linalg.norm((y[i,:]-h[l]))
    else:
        if l > 1:
            delta[l] = calculate_hidden_delta(delta[l+1], W[l], z[l])
            #  $\text{tri}W^{(l)} = \text{tri}W^{(l)} + \text{delta}^{(l+1)} * \text{transpose}(h^{(l)})$ 
            tri_W[l] += np.dot(delta[l+1][:,np.newaxis], np.transpose(h[l][:,np.newaxis]))
            #  $\text{trib}^{(l)} = \text{trib}^{(l)} + \text{delta}^{(l+1)}$ 
            tri_b[l] += delta[l+1]
# perform the gradient descent step for the weights in each layer
for l in range(len(nn_structure) - 1, 0, -1):
    W[l] += -alpha * (1.0/m * tri_W[l])
    b[l] += -alpha * (1.0/m * tri_b[l])
# complete the average cost calculation
avg_cost = 1.0/m * avg_cost
avg_cost_func.append(avg_cost)
cnt += 1
return W, b, avg_cost_func

```

```

def predict_y(W, b, X, n_layers):
    m = X.shape[0]
    y = np.zeros((m,))
    for i in range(m):
        h, z = feed_forward(X[i, :], W, b)
        y[i] = np.argmax(h[n_layers])
    return y

```

```

if __name__ == "__main__":
    # load data and scale
    digits = load_digits()
    X_scale = StandardScaler()
    X = X_scale.fit_transform(digits.data)
    y = digits.target
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4)
    # convert digits to vectors
    y_v_train = convert_y_to_vect(y_train)
    y_v_test = convert_y_to_vect(y_test)
    # setup the NN structure
    nn_structure = [64, 30, 10]
    # train the NN
    W, b, avg_cost_func = train_nn(nn_structure, X_train, y_v_train)
    # plot the avg_cost_func
    plt.plot(avg_cost_func)
    plt.ylabel('Average J')

```

```
plt.xlabel('Iteration number')
plt.show()
# get the prediction accuracy and print
y_pred = predict_y(W, b, X_test, 3)
print('Prediction accuracy is {}'.format(accuracy_score(y_test, y_pred) * 100))
```

Додаток В

Специфікація

ГЮИК. 506190.007

Додаток Г

Відомість атестаційної роботи

ГЮИК. 506190.007 ДЗ

(позначення документу)

