

# Прогнозування часу виконання операцій множення матриць на графічному процесорі



**Аналіз даних**

	MWG	NWG	KWG	MDIMC	NDIMC	MDIMA	NDIMB	KWI	VWM	VWN	STRM	STRN	SA	SB	Run1 (ms)
0	16	32	16	8	16	8.0	8.0	8.0	2.0	2.0	1.0	1.0	1.0	1.0	39.14
1	128	32	32	32	8	16.0	8.0	8.0	4.0	4.0	0.0	0.0	1.0	1.0	32.25
2	128	32	32	16	32	32.0	16.0	8.0	2.0	1.0	0.0	0.0	1.0	0.0	60.95
3	128	64	32	32	16	16.0	32.0	2.0	1.0	1.0	0.0	1.0	0.0	0.0	132.85
4	64	16	32	16	8	8.0	8.0	2.0	2.0	1.0	1.0	1.0	0.0	1.0	47.13
---															
<b>101211</b>	128	128	32	16	16	16.0	8.0	2.0	1.0	8.0	1.0	1.0	1.0	0.0	141.61
<b>101212</b>	64	64	32	32	8	8.0	16.0	8.0	1.0	1.0	1.0	1.0	0.0	0.0	56.15
<b>101213</b>	64	128	16	8	8	16.0	32.0	8.0	4.0	2.0	0.0	1.0	1.0	1.0	874.63
<b>101214</b>	128	64	16	16	8	16.0	8.0	2.0	1.0	4.0	0.0	0.0	0.0	1.0	233.63
<b>101215</b>	16	64	32	16	16	16.0	8.0	2.0	1.0	1.0	1.0	0.0	1.0	0.0	89.19

101216 rows × 15 columns

**MWG:** the matrix width and height

**NWG:** the matrix width and height

**KWG:** the internal dimension of the matrix block

**MDIMC:** the size of the thread block for matrix multiplication

**NDIMC:** the size of the thread block for matrix multiplication

**MDIMA:** the size of the thread block for matrix addition

**NDIMB:** the size of the thread block for matrix addition

**KWI:** the kernel loop unrolling factor

**VWM:** the matrix read block width

**VWN:** the matrix read block width

**STRM:** enable stride for accessing matrix

**STRN:** enable stride for accessing matrix

**SA:** the matrix A type

**SB:** the matrix B type

# Формування навчального та тестового датасету

```
df = pd.read_csv("sgemm_product.csv")
df = shuffle(df)
df_test = df[-20000:]
df = df[:80000]
```

**Лінійна регресія**

# МІНІМІЗАЦІЯ КОЕФІЦІЕНТІВ

$$I(a) = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2 \underset{a}{\rightarrow} \min$$

$$y_i = a_0 + a_1 x_i^{(1)} + \dots + a_k x_i^{(k)}$$

# L1 - регуляризація

$$I(a) = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2 + \mu \sum_{j=1}^k |a_j| \rightarrow \min_a$$

```
if reg == "L1":  
    gradient += u
```

# L2 - регуляризація

$$I(a) = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2 + \mu \sum_{j=1}^k a_j^2 \rightarrow \min_a$$

```
if reg == "L2":  
    gradient += coef * 2 * u
```

# МІНІМІЗАЦІЯ КОЕФІЦІЕНТІВ методом градієнтного спуску

$$a^{(0)} = (a_0, a_1, \dots, a_k)$$

$$a^{(k+1)} = a^{(k)} - \beta \cdot \nabla I(a^{(k)}, x, y)$$

$$|a^{(k+1)} - a^{(k)}| < \varepsilon$$

`coef_new = coef - learning_rate * gradient`

**Результаты работы**

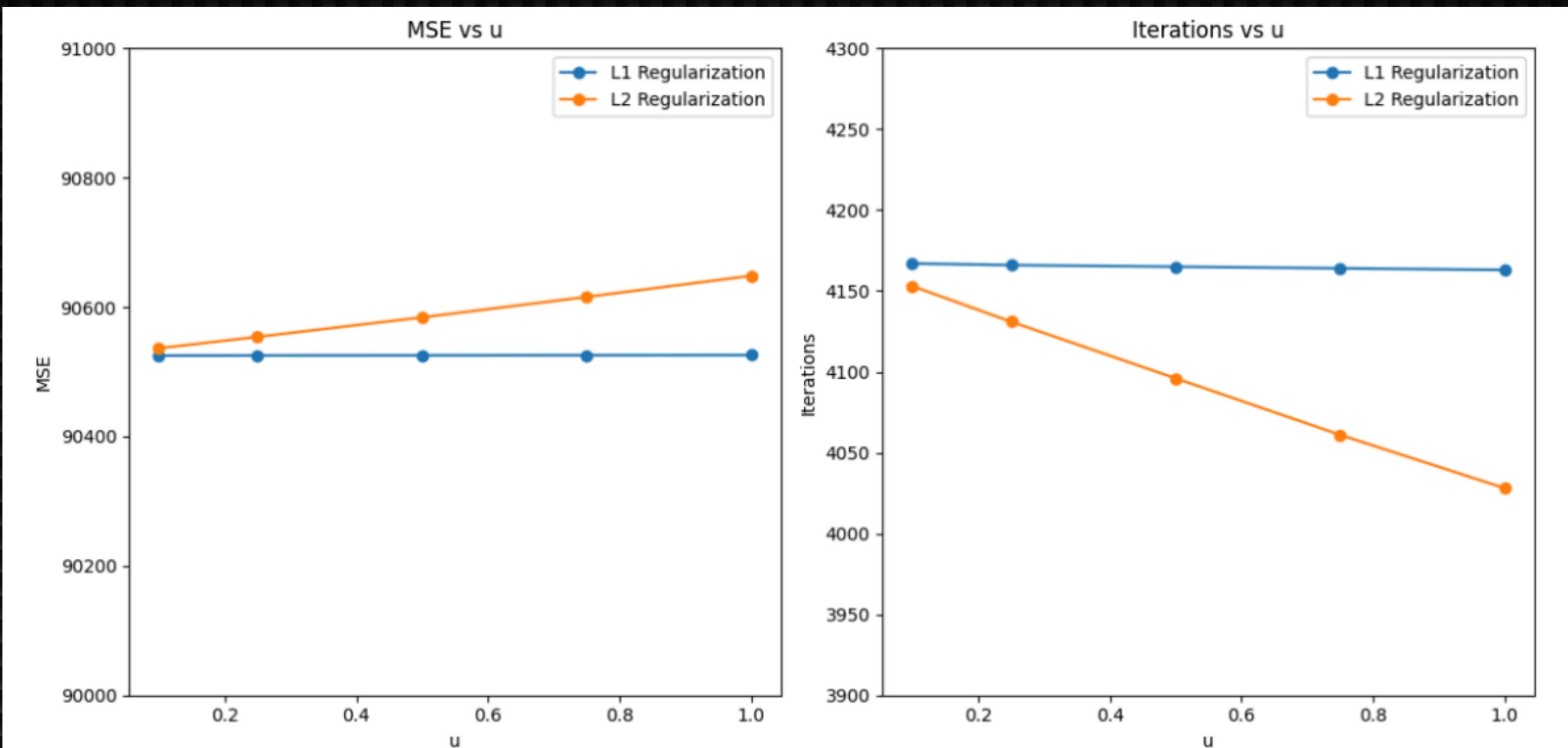
Reg: no, lr: 1e-05

Number of iterations: 4167

Time taken (seconds): 40.698108434677124

Test Mean Squared Error (MSE): 90525.18544948414

=====



# Висновки