國立中央大學機械工程學系
Department of Mechanical Engineering, National Central University

110學年大學部畢業專題海報成果展


賴柏綸　郭洧銓　連爾諾　黃曜駿

FESTO產線排程優化技術

作者：賴柏綸、黃曜駿、連爾諾、郭洧銓
指導教授：林錦德

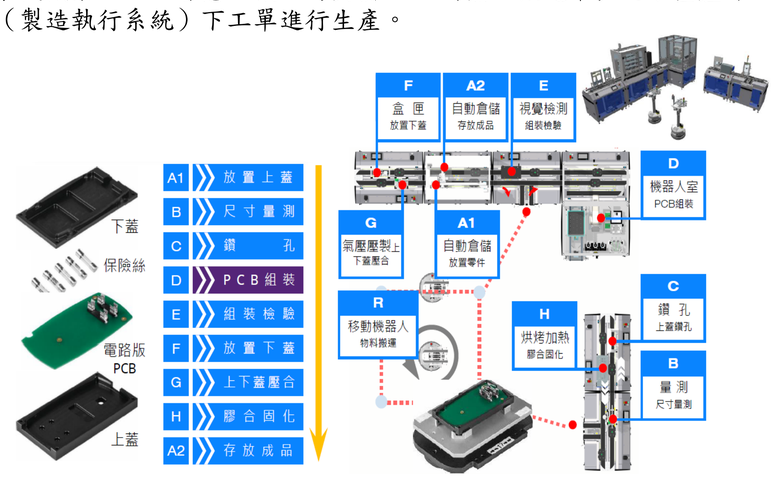
編號 19

簡介

傳統的排程預測技術僅適用固定標準加工時間的流程，然而有些機台會因為環境因素影響，導致沒有一個固定標準工時（如：加熱製程會隨著環境溫度而導致加熱時間長度不同），因此我們引入深度學習技術幫助我們預測這些非固定的標準工時，並結合排程技術縮小傳統排程的誤差，使排程系統能有效最佳化產線加工時間。

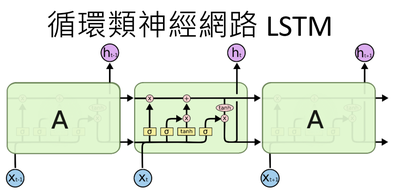
實作場域：

中央大學機械系FESTO智慧工廠。為具備AGV（無人搬運車）之小型產線，可以透過MES（製造執行系統）下工單進行生產。



研究方法

以深度學習(遞迴類神經網路LSTM)結合排程系統(OR-Tools)優化產線流程，並透過 OPC UA協定控制FESTO工廠，以有效優化FESTO工廠產線加工工時。



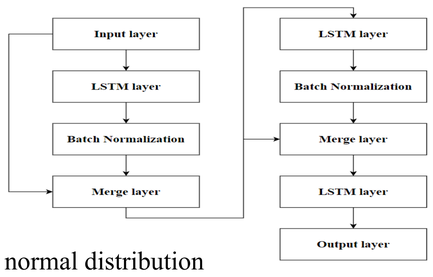
深度學習模型

因各工單的加熱時間不同，所以利用Python建立循環類神經網路(LSTM)，並利用收集之資料來訓練模型，並以此模型來預測工單在加熱站之加熱時間。

輸入：[當前加熱機台內溫度 , 加熱機台是否在加熱]

輸出：[下一秒機台溫度]

模型架構如下：



LSTM層設定如下：

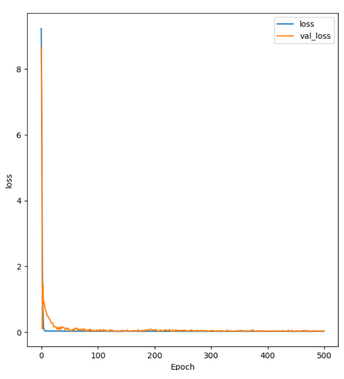
權重初始化：Glorot normal distribution

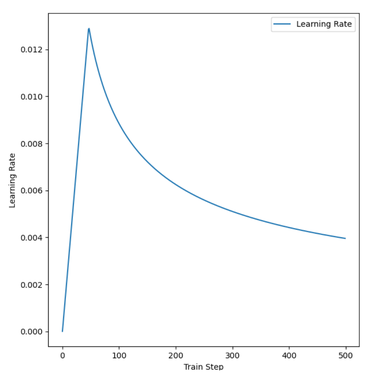
Loss Function：Mean Absolute Error並加上Ridge regression限制權重大小

Optimizer：Adam $\beta_1 = 0.9$，$\beta_2 = 0.98$,$\epsilon = 1e-9$

Learning rate：使用Warm Up方法（如下圖）

最後訓練完loss與Learning rate如右圖：

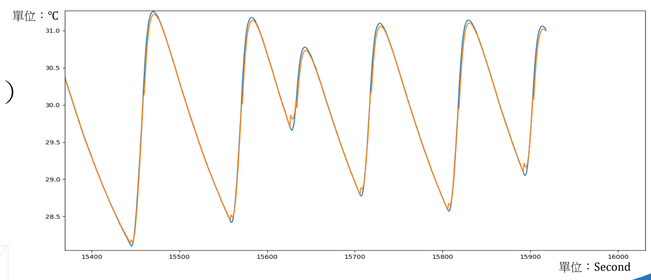




最後測試集對比結果：

（藍色：實際溫度值）

（橘色：LSTM預測值）



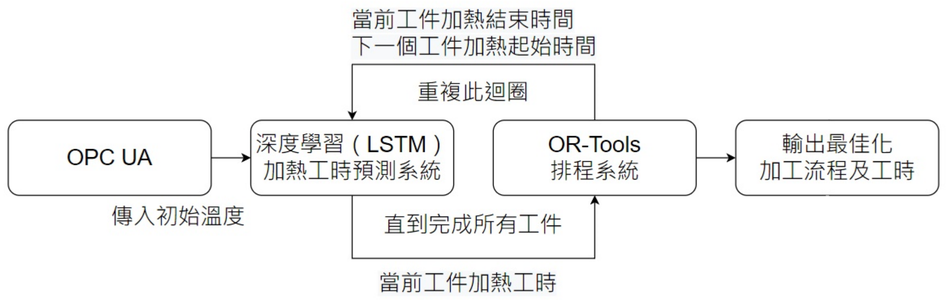
排程系統

我們使用Google開發的開源函式庫OR-Tools來撰寫排程系統，在規劃FESTO工廠的排程問題時，我們遇到了許多Constraint Programming問題，OR-Tools 函式庫能夠有效的幫助我們解決問題。在撰寫排程系統中，我們將工廠許多Constraint Problem編寫進排程系統內（如：AGV派送運輸問題，輸送帶運輸時間等），並使用OR-Tools的優化器，計算出最佳的工時及加工流程。

```
# Create and add disjunctive constraints.
for machine in all_machines:
    self.__model.AddNoOverlap(machine_to_intervals[machine])
# Precedences inside a job.
for job_id, job in enumerate(self.jobs_data):
    for task_id in range(len(job) - 1):
        # model.Add(all_tasks[job_id, task_id +1].start >= all_tasks[job_id, task_id].end)
        """限制條件: 各站運輸時間(輸送帶運送時間)"""
        self.__model.Add(all_tasks[job_id, task_id +1].start >= all_tasks[job_id, task_id].end + self.P2P_time[task_id])
for i in  range(len(self.jobs_data)):
    if (i != 0):
        self.__model.Add(all_tasks[i, 1].start >= all_tasks[0, 1].end + 46)
        self.__model.Add(all_tasks[i, 1].start >= all_tasks[i-1, 1].end + 46)
        self.__model.Add(all_tasks[i, 9].start >= all_tasks[i-1, 9].end + 46)
        self.__model.Add(all_tasks[i, 11].start >= all_tasks[i-1, 11].end + 46)
        self.__model.Add(all_tasks[0, 9].start >= all_tasks[i, 1].end + 46)
# Makespan objective.
obj_var = self.__model.NewIntVar(0, horizon, 'makespan')
self.__model.AddMaxEquality(obj_var, [
    all_tasks[job_id, len(job) - 1].end
    for job_id, job in enumerate(self.jobs_data)
])
self.__model.Minimize(obj_var)
```

深度學習結合排程系統

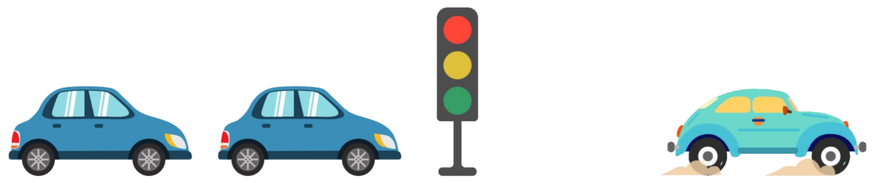
這個專題最有挑戰的地方是我們如何將深度學習運用在排程系統上，前面提到過我們深度學習是用來預測固定的標準工時，我們訓練的LSTM模型可以精準的預測每秒加熱機台內的溫度，我們只需要透過OPC UA得知機台初始的溫度，提供給深度學習系統計算出加熱的時間後，排程系統會計算出當前工件的工作時間，並且同時會將當前工件加熱完畢時間以及下一個工件的加熱起始時間，再交給深度學習系統計算出加熱時間，重複以上動作直到計算完所有機台的加熱時間，最後排程系統會將所有資料一併計算出最後的優化結果。



OPC UA產線控制

OPC UA (OPC Unified Architecture) 為自動化技術的機器對機器網路傳輸協定，透過此協定可以得知各工作站的運行狀態、各項控制器開關、讀取感應器等等，甚至可以連線至機台中，更改機台的控制參數，進而實現控制加工流程的目的。

我們運用此協定撰寫機台控制系統，以Python作為開發語言。機台控制系統可以讀取各機台參數資料，也可以更改機台內部的參數，用以達成我們控制的目的。依據排程預測加工流程結果，我們在輸送帶加入紅綠燈的概念，控制載具順序，進而改變AGV的運行規律、控制機台以達成排程系統預測之加工流程。



結論

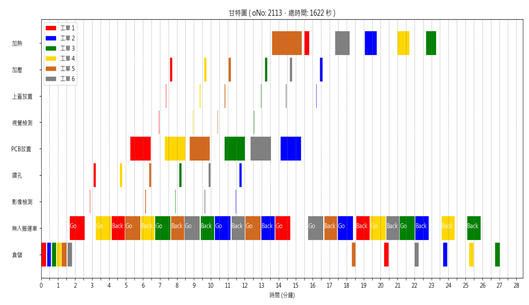
經由產線控制實現排程結果後，可以有效節省以下幾種時間浪費：

(1) AGV因無法預測工單進入AGV等待區的所需時間而造成AGV多跑一趟

(2) 工單因AGV等待區無空閒造成空跑圈

我們以六單的甘特圖為例，優化後的工時減少超過14%。

未優化結果(六單，時間：1622秒)



優化後結果(六單，時間：1391秒)