

Student: Jonathan Benitez
Handledare: Christian Antfolk
Examinator: Johan Nilsson

Populärvetenskaplig sammanfattning av examensarbete:

Implementation och utvärdering av maskininlärningsalgoritmer i en mikrokontroller för kontrollsystem till myoelektrisk protes

Till folk som har övre lemmamputering så finns det många olika proteser. Något som håller på att utvecklas nu är proteser som kontrolleras av muskelkontraktioner, även kallad myoelektrisk protes. Myoelektrisk protes fungerar genom att ha sensorer som mäter den elektriska aktiviteten i muskler, dessa sensorer placeras på muskler och metoden kallas surface Electromyography (sEMG). Från sEMG mätningen så används Artificiell Intelligens (AI) för att klassificera vilken elektrisk mätning motsvarar vilken rörelse. Det finns olika AI modeller som kan klassificera rörelser från sEMG data och de olika modellerna är bra på olika sätt. Att träna en AI modell kräver mycket beräkningskraft, men när den är tränad så kan den implementeras till en mindre dator, även kallad mikrokontroller. Att använda en mikrokontroller är billigare och förbrukar mindre energi. Under det här examensarbetet skrevs det ett program som genererar kod för några olika AI modeller som en mikrokontroller sedan kan använda för att klassificera olika rörelser. När implementeringen var klar utforskades även de olika begränsningarna programmet hade.

Implementering av programmet delades upp i 3 steg: samla data i ett fönster, ta ut "gömd" information från detta fönster, använd den "gömda" informationen som insignal till en AI modell för att klassificera rörelsen. För att extrahera den "gömda" informationen så finns det olika algoritmer, tolv stycken av dessa extraheringsalgoritmer implementeras och testades. Tre olika typer av AI modeller kunde implementeras med hjälp av programmet: Linjära modeller, Neurala Nätverk (NN) och Convolution Neurala Nätverk (CNN). De linjära modeller är modeller där man separerar datapunkter genom att dra linjer. Genom att se mellan vilka linjer ett datavärde är så kan datapunkten klassificeras till en rörelse. NN försöker efterlikna hur neuroner fungerar i en människa, neuronerna är kopplade i lager och deras utsignal beror på värdena från neuronerna i tidigare lager. CNN använder fönsterdata i stället för den extraherade informationen från fönsterdatan. Detta är eftersom CNN ser fönsterdata som en bild, och från denna bilden så kan den hitta mönster och därifrån veta vilka mönster leder till vilken rörelse.

När programmet implementerats så utfördes olika tester, där det första var att testa de tolv olika extraheringsalgoritmerna. Resultatet visade att det inte fanns en klar vinnare bland algoritmerna och vissa algoritmer var bättre för några människor och andra algoritmer var bättre för andra människor. Detta visar att det finns en poäng med att använda det implementerade programmet, så att varje person får en modell som fungerar bäst för en. De allra flesta Linjära modeller kunde implementeras utan problem, för NN så uppkom det minnesproblem för större modeller och för CNN uppkom det även minnesproblem för större modeller men även tidsproblem för några specifika modeller. Det fanns alltid en liten förlust för AI modellerna när de implementeras i mikrokontrollern, men var så pass små att de borde kunna försummas. I framtiden så skulle programmet kunna utvecklas för att kunna implementera fler extraheringsalgoritmer och fler AI modeller. Programmet skulle även kunna optimeras så att den genererade koden skulle ha färre minnes- och tidsproblem.