

EXAMENSARBETE Type Inference in PHP using Deep Learning**STUDENTER** Samuel Klingström, Pontus Olsson**HANDLEDARE** Niklas Fors (LTH)**EXAMINATOR** Görel Hedin (LTH)

Kan maskininlärning användas för typinferens?

POPULÄRVETENSKAPLIG SAMMANFATTNING Samuel Klingström, Pontus Olsson

Stöd för typpannoteringar i dynamiska programmeringsspråk blir allt vanligare, men att uppdatera äldre kod med typpannoteringar kan vara ett väldigt tidskrävande jobb. Detta arbete använder maskininlärning för att gissa typer på funktioner och metoder.

Dynamiska programmeringsspråk som t.ex PHP, Python, och JavaScript har under de senaste åren börjat stödja typpannoteringar likt statiskt typade programmeringsspråk som t.ex Java och C++. För tillfället finns det många verktyg som kan analysera kod för att hitta fel, men för PHP är det få av verktygen som klarar av att inferera typer.

I vårt examensarbete har vi skapat ett verktyg som använder maskininlärning för att gissa typer på parametrar i funktioner och metoder för PHP. Maskininlärningsmodellen kan, givet en funktion eller metod, gissa vilka typer som hade passat varje parameter baserat på hur den används. För varje parameter rapporterar verktyget sina tre bästa gissningarna direkt i kodkommentaren, samt hur säker modellen är på varje gissning. Ett exempel visas figuren där nätverket gissar typen `int` för parametern `$n`.

Genom att ge förslag på typpannoteringar till utvecklare, blir det mycket enklare för utvecklare att uppdatera kod som inte har typpannoteringar. Att uppdatera gammal kod som saknar typpannoteringar är en väldigt tidskrävande process, då utvecklare måste sätta sig in i hur koden fungerar för att kunna uppdatera koden korrekt. Ett verktyg som detta underlättar processen, och utvecklare kan istället spendera mer tid på att skapa ny funktionalitet.

```
1 /**
2  * Predictions for $n
3  * 1. int 99.92 %
4  * 2. float 0.07 %
5  * 3. double 0.0 %
6  */
7 function factorial($n) {
8     $factorial = 1;
9     while ($n > 0) {
10         $factorial = $factorial * $n;
11         $n--;
12     }
13     return $factorial;
14 }
```

För att modellen ska kunna göra gissningar på kod, måste koden representeras på ett sätt som modellen förstår. För varje fil som ska användas läses koden in och omvandlas till ett träd, där varje nod i trädet representerar en del av strukturen hos koden. Utifrån trädet kan sen information kring koden fås genom att titta på hur de olika noderna i trädet är kopplade till varandra.

Efter att modellen tränats med de 10 000 mest populära PHP-projekten på Github, kunde den gissa typpannoteringar med en korrekthet på 75 % om den fick en gissning, och 84 % om den fick tre gissningar. Dessa resultat är betydligt bättre än vad nuvarande statiska kodanalysverktyg presterar. Att använda maskininlärning för typinferens ser därför ut att vara väldigt lovande.