**Hard-to-Predict (H2P) σύφμωνα με το paper:**

Σύμφωνα με το Paper (H2P analysis.pdf):

1. Τρέχουμε τα benchmarks με διαφορετικα workloads (κάθε benchmark δέχεται ενα input i.e. workload, π.χ. το 602.gcc\_s κάνει compile C κώδικα, τα input είναι διαφορετικά C source κώδικες).
2. Για κάθε trace, ανα 30M εντολές, εντοπίζουμε branches που εμφανίζονται τουλάχιστον 15K φορές, δημιουργούν τουλάχιστον 1000 misspredictions και έχουν accuracy λίγοτερο απο 99% κάτω από τον Tage 8ΚΒ predictor. Τα κριτήρια επιλέγονται με τη λογική: πολλά misspredicts (άρα περιθώρια βελτείωσεις) και πολλά occurrences άρα κατάληλλα για machine learning προσέγγιση.
3. Βλέπουμε ότι αν γίνει σωστή πρόβλεψη των συγκεκριμένων branches έχουμε μια αύξηση στο IPC, ειδικά όταν κάνουμε scale up το pipeline (reorder buffer size, fetch width, execution width, load/store queues) το IPC αυξάνεται ακόμα περισσότερο.

Στη συνέχεια στο paper “CNN for H2P.pdf”:

1. Δημιουργεί CNN νευρωνικό το οποίο δέχεται input πίνακα 200x256. Κάθε γραμμή απο τις 200 συμβολίζει με τη σειρά τα προηγούμενα αποτελέσματα των branches που έχουν παρέλθει. Κάθε γραμμή είναι γεμάτη με 0 και έχει έναν και μόνο άσσο σε μία απο τις 256 θέσεις σύμφωνα με: Χρησιμοποιούμε τα 7 LSB του Program Counter (PC), κάνουμε 1 logic shift left και προσθέτουμε 0 ή 1 ανάλογα με το αποτέλεσμα του συγκεκριμένου branch. Για παράδειγμα αν θέλουμε να κωδικοποιήσουμε το 10 στη σειρά branch με PC 8325735 και αποτέλεσμα True, τότε στην 10η σειρά του πίνακα θα γεμίσουμε μηδενικά. Για να βρούμε τη θέσου του άσσου αρκεί να πάρουμε τα 7LSB του PC, δηλαδή τον δεκαδικό αριθμό 103, να κάνουμε shift logic left, άρα γίνεται 206 και να προσθέσουμε 1, αφού το αποτέλεσμα ήταν True, άρα στη θέση 207 βάζουμε άσσο.
2. Στο paper βλέπουμε στα αποτελέσματα ότι κάνοντας train για κάποιο συγκεκριμένο benchmark και workload, το νευρωνικό αποδίδει καλά για οποιοδήποτε άλλο workload του ίδιου benchmark. Συνεπώς μπορούμε να φτιάξουμε κάποια αρχιτεκτονική η οποία αποδίδει καλά σε στοχευμένα προβλήματα το οποίο μπορεί να αξιοποιηθεί π.χ. σε κάποιο datacenter που επιτελούν συγκεκριμένες εργασίες.

**Παρατηρήσεις:**

1. Δεν έχουμε πρόσβαση στα benchmarks ώστε να μπορέσουμε να δοκιμάσουμε με διαφορετικά workloads. Ο μόνος τρόπος για να προσεγγίσουμε το ζήτημα, είναι να κάνουμε Train σε κάποιο simpoint και επαληθεύοντας την απόδοση του νευρωνικού σε κάποιο άλλο simpoint.
2. Γενικά όταν αντιμετωπίζουμε κάποιο machine learning πρόβλημα, ξεκινάμε επιλέγοντας μοντέλο που έχει αποδόσει σε παρόμοια προβλήματα. Το δικό μας πρόβλημα φαίνεται να είναι ένα Sequence Classification πρόβλημα, στο οποίο συνήθως αποδίδουν RNN νευρωνικά. Τα CNN που χρησιμοποιούν στο paper φαίνεται να αποδίδουν καλύτερα σε spatial-Data προβλήματα (χωρικά) όπως π.χ. όταν έχουμε να κάνουμε με εικόνες. (Για αυτό και το input στο paper είναι ένας ολόκληρος πίνακας 200χ256. Αντίθετα σε ένα LSTM νευρωνικό το input θα ήταν ένας μονοδιάστατος πίνακας 256 θέσεων, και θα δινόνταν το ιστορικό ένα-ένα τη φορά, σαν το διάβασμα μίας πρότασης 200 λέξεων). Δεν αναφέρονται οι λόγοι που προτίμησαν ένα CNN ωστόσο πιθανότοτα η επιλογή έγινε είτε γιατί δοκιμάστηκε και το RNN αλλά δεν απέδωσε, είτε επειδή ο τελικός σκοπός του νευρωνικού είναι να ενσωματωθεί στο hardware, πράγμα πιο εύκολο για ένα CNN.